

Taxometric Programs for the R Computing Environment: User's Manual

John Ruscio, The College of New Jersey (ruscio@tcnj.edu)

TABLE OF CONTENTS

1 Overview	7 MAMBAC – Performing Mean Above Minus Below A Cut (MAMBAC) Taxometric Analyses
2 Getting Started in the R Environment	7.1 Overview
2.1 Overview	7.2 Command
2.2 Accessing the Taxometric Programs	7.3 Required Arguments
2.3 Viewing and Removing Objects	7.4 Optional Arguments
2.4 Importing Data Files	7.5 Running the Program
2.5 Saving and Retrieving a Workspace	7.6 Output
2.6 Getting Acquainted with R	8 MAXEIG – Performing Maximum Covariance (MAXCOV) or Maximum Eigenvalue (MAXEIG) Taxometric Analyses
2.7 Adapting the Programs for Mac Use	8.1 Overview
3 Createdata – Generating Artificial Categorical or Dimensional Data Sets	8.2 Command
3.1 Overview	8.3 Required Arguments
3.2 Command	8.4 Optional Arguments
3.3 Required Arguments	8.5 Running the Program
3.4 Optional Arguments	8.6 Output
3.5 Running the Program	9 L-Mode – Performing Latent Mode (L-Mode) Taxometric Analyses
3.6 Output	9.1 Overview
4 GenData – Generating Comparison Data Sets	9.2 Command
4.1 Overview	9.3 Required Arguments
4.2 Using Comparison Data in Taxometric Studies	9.4 Optional Arguments
4.3 An Objective Index of Curve Fit	9.5 Running the Program
4.4 Differences Between DimSample and SimDim	9.6 Output
4.5 Further Modifications of DimSample	10 MAXSLOPE – Performing Maximum Slope (MAXSLOPE) Taxometric Analyses
4.6 GenData Replaces DimSample and TaxSample	10.1 Overview
5 Indicator.Dist – Describing Indicator Distributions, Correlations, and Validity for a Data Set	10.2 Command
5.1 Overview	10.3 Required Arguments
5.2 Command	10.4 Optional Arguments
5.3 Required Arguments	10.5 Running the Program
5.4 Running the Program	10.6 Output
5.5 Output	11 P.Classify – Implementing the Base-Rate Classification Technique
6 Skew & Kurtosis – Calculating the Skew and Kurtosis of a Distribution	11.1 Overview
6.1 Overview	11.2 Command
6.2 Commands	11.3 Required Arguments
6.3 Required Arguments	11.4 Optional Arguments
6.4 Running the Programs	11.5 Running the Program
6.5 Output	11.6 Output
	12 Program Updates & References – History of Updates to the Program Code and References Cited in this Manual

Chapter 1: Overview

The accompanying file (**TaxProg [date].R**) contains a suite of programs written in the R language for use in a taxometric investigation. These programs were originally written in the S+ language, and then converted to R on October 25, 2002. R is available as a free download from any of the sites listed at <http://cran.r-project.org/mirrors.html>. I'm currently using R version 3.1.1 (released 7/10/2014) on a Mac, though the program code may run on other versions of R or on other platforms. A brief overview of how to get started using R (including some notes on how to adapt the programs for Mac use) is provided on the next page, followed by detailed descriptions of the following functions that are currently included in the accompanying taxometrics program file:

- *CreateData* generates artificial categorical or dimensional data according to a handful of basic parameters (e.g., sample size, base rates, number of indicator variables, indicator validity).
- *GenData* generates comparison data sets that reproduce the distributional and correlational characteristics of a supplied set of data. This program replaces *DimSample* and *TaxSample*, which replaced *SimDim* and *SimTax*.
- *Indicator.Dist* provides summary statistics on indicator distributions and correlations (full-sample and within-group) as well as between-group indicator validity. This program replaces the older *Validity.Est*.
- *Skew* and *Kurtosis* are functions to compute these descriptive statistics, which are not otherwise available in R. This information is provided by *Indicator.Dist*, but one can use these stand-alone programs for more circumscribed purposes.
- *P.Classify* assigns cases to groups using the base-rate classification technique.
- *MAMBAC*, *MAXEIG*, *MAXSLOPE*, and *LMode* implement the familiar taxometric procedures of the same names. The *MAXEIG* program can be used to perform *MAXCOV* or *MAXEIG* analyses.

Some of the features of this suite of programs include the following:

- For each taxometric procedure, program parameters allow for considerable flexibility in the ways that indicator variables are utilized and analyses are conducted.
- Whenever possible, results can be pooled across internal replications to increase the interpretability of taxometric graphs that might otherwise be affected by locating thresholds arbitrarily between equal-scoring cases.
- The taxometric programs offer automated generation and analysis of categorical and dimensional comparison data and provide an objective index of the extent to which the curves produced by comparison data fit those of the research data (the CCFI). See Ruscio, Walters, Marcus, and Kaczetow (2010) for suggestions on using the CCFI for consistency testing.
- Graphical output is placed into graph sheets labeled according to the type of analysis. Text output includes data parameters, program specifications, and statistical information; this can be redirected to a file.
- A number of parameters allow control over the graphing of results. Some features can render the taxometric curves more interpretable, whereas others assist in the organization of results. All programs yield fully and clearly labeled taxometric graphs or panels of graphs. Reorganizing the graphs for presentation may require saving them for editing using other software; they can be copied-and-pasted or printed, as is, straight from R. The “metafile” (.emf) format retains detail in a relatively small file size.

The detailed documentation that follows describes the required and optional parameters of each program as well as the output that is produced. In addition, important decisions that you must make in setting up the analyses are discussed along with strategies for using these programs to do so. A basic knowledge of taxometrics is presumed. I would recommend reading through all of the available options and related discussions to get a sense for what the programs are capable of and to make thoughtful choices about how best to conduct your analyses. *Relying on default settings may be inappropriate in many circumstances.* Feel free to contact me with questions or comments about the use of these programs, to report bugs, or to suggest additional features that would be useful.

The final chapter of this manual contains a listing of updates to the program code by date. You can consult this information to see what changes, if any, have been made since you last downloaded a copy of the program code. References are provided here as well.

Chapter 2: Getting Started in the R Environment

2.1 Overview:

The R computing environment contains a wide array of mathematical, statistical, and graphing tools, and it runs fairly quickly compared with programs such as S+. A handful of commands can be accessed via pull-down menus, but most require entering text commands.

R organizes information using a “Console” window for entering commands and displaying text results plus additional windows that are created as needed for presenting graphical results. Text output that appears in the console window can be copied and pasted into other files or applications, and output from taxometric programs can be redirected to a file. Graphical output will appear in separate graph sheets (one page of graphs per graph sheet) and can either be printed as-is (this is how all of the graphical samples shown in this manual were created) or saved as a graphics file for editing or use in other software. For example, you can save a graph as a jpeg file and include it in a Word document. Unfortunately, there is no interactive utility to allow you to modify elements of a graph or reorganize graphs within a graph sheet. If you want your axes labeled or scaled differently, for example, you would have to edit the program code.

R is object-oriented and allows users to store multiple programs and/or data objects within a single “workspace”. Thus, rather than having a large number of data files that are associated with a single study, one can save and retrieve them all in a single R workspace.

R is case sensitive. For example, whereas a command beginning `MAMBAC` will run the appropriate taxometric program, a command beginning `Mambac` will not be recognized. The command prompt in R is the symbol “>”; in what follows, any lines that display commands begin with this prompt and appear in bold print to distinguish them from the text output that results.

2.2 Accessing the Taxometric Programs:

When you first start an R session using the graphical user interface (Rgui), you can load the taxometric programs by choosing the **Source File (or Source R Code on a PC)** option from the **File** menu and opening `TaxProg.R` (with the actual file name containing the date of the program file, e.g. `TaxProg 2014-07-29.R`).

2.3 Viewing and Removing Objects

To verify that the taxometric programs have been entered into R properly, type `objects()`. This command lists all available objects, including programs and data files. For example, once the taxometric programs have been accessed but before you have imported or created any data objects, you should see the following list of 28 objects:

```
> objects()
```

[1] "CreateData"	"Factor.Analysis"	"Factor.Analysis.LMode"	"Fit.Densities"
[5] "GenData"	"Indicator.Dist"	"Kurtosis"	"LMode"
[9] "LMode.Main"	"MAMBAC"	"MAMBAC.Calculate"	"MAMBAC.Main"
[13] "MAMBAC.P.Est"	"MAMBAC.Plot"	"MAMBAC.Structure"	"MAMBAC.Summarize"
[17] "MAXEIG"	"MAXEIG.Calculate"	"MAXEIG.Main"	"MAXEIG.P.Est"
[21] "MAXEIG.Plot"	"MAXEIG.Structure"	"MAXEIG.Summarize"	"MAXSLOPE"
[25] "MAXSLOPE.Main"	"P.Classify"	"Remove.Missing"	"Skew"

You can delete an object using the `rm()` command, with the name(s) or one or more objects to be removed listed in the parentheses. For example, if a command leads to a large number of warning messages, R places them all within an object calling `last.warning`; to delete this object, you would type `rm(last.warning)`.

2.4 Importing Data Files:

To import data (e.g., from SPSS or SAS files), there is a supplementary package of R functions. Type `library(foreign)` to load these, and then from the **Help** menu, choose **Manuals** and then **R Data Import/Export** to read about them. Additional help is available by typing the name of any R command in quotes and parentheses after the word “help.” For example, after loading the import/export commands, you could type `help("read.spss")` to see how to import data from an SPSS file.

2.5 Saving and Retrieving a Workspace:

As noted above, R allows you to save an entire workspace as a single file. A workspace consists of a set of objects, and you can create separate workspaces for each research project that that will include all of the program and data objects created during work on that project. When you restart another R session, you can do so via an R icon if you created one, by accessing R from a menu, or by opening a saved workspace file (one that ends in .RData); this latter method starts R and loads the chosen workspace. If you have already started an R session, you can retrieve a saved workspace. Note that if you have already opened one workspace, you can add to it the contents of another by opening the latter; new objects will be added to the current workspace, and objects with the same name will overwrite those in the current workspace.

In my experience, newer versions of R appear to crash much less frequently than older versions. Nonetheless, if you are making progress during an R session I would recommend saving your output and/or workspace frequently. If you manually exit the program, you will be prompted to save the workspace; doing so saves it as the current workspace that will automatically be loaded when you next start an R session, *not* using whatever file name you might wish to specify for that project.

2.6 Getting Acquainted with R:

Beyond these basics, I can provide little help with the R language itself. There is quite a bit of documentation included with the full R package available as a free download. With the taxometric programs loaded and some test data available, you can play with any of the programs described in subsequent chapters. I would recommend at least skimming through this manual before doing so, or (even better) typing the sample commands as you read through this to see how the results are presented and compare them to what is shown here. One final note on entering the commands: I have included default settings for many parameters of my programs, and you only need to type elements of the command that (1) are required, with no default available (such as supplying the name of the object that contains your data), or (2) request non-default settings. The sample commands in this documentation illustrate the use of default settings as well as the occasional specification of alternative settings.

2.7 Adapting the Programs for Mac Use:

I am grateful to two individuals who have offered helpful advice. David Strong writes that to modify the code to work on the 'RAqua' version for Mac OS X, you should replace each instance of `dev.new()` with `quartz(display="0:0")`. Harold Kincaid writes that if this does not work, you can instead add the following lines at the beginning of the main program file:

```
if(.Platform$OS.type=="windows") {  
  quartz<-function() windows()  
}
```

These lines were added to the program code on 2/26/2014, so Mac users should not need to modify the file themselves.

Chapter 3: CreateData

Generating Artificial Categorical or Dimensional Data Sets

3.1 Overview:

This program creates an artificial data set based on either dimensional or categorical latent structure, including within-group correlations, skew, and/or ordered categorical values if desired. The program returns a data object containing the indicators plus, if categorical, a final column containing group membership (1 = complement, 2 = taxon). Such data can be useful for getting to know the taxometric programs and becoming familiar with their output by conducting analyses using data sets whose parameters are known.

3.2 Command:

```
CreateData <- function(Str, N = 600, k = 4, P = .50, d = 2.00, r = .00, Tax.r = .00, Comp.r = .00, Skew = 0,
                      Cuts = 0, Seed = 1)
```

3.3 Required Arguments:

`Str` – Type of data to be generated. Use "Dim" for dimensional, "Cat" (or anything other than "Dim") for categorical.

3.4 Optional Arguments:

`N` – Number of cases (default = 600).

`k` – Number of indicator variables (default = 4).

`P` – Taxon base rate (default = .50).

`d` – Validity with which each indicator separates groups, in within-group SD units (default = 2.00). A single averaged value is specified that will be used for all indicators.

`r` – Correlation between indicators (default = .00). This is used when generating dimensional data. If it is not specified, the program calculates a correlational equivalent of the specified values of `P`, `d`, `Tax.r`, and `Comp.r` (see formula in Meehl & Yonce, 1994, p. 1146). A single average correlation is specified.

`Tax.r` – Within-group correlations, or "nuisance covariance," for taxon members (default = .00). A single average value is specified.

`Comp.r` – Within-group correlations, or "nuisance covariance," for complement members (default = .00). A single average value is specified.

`Skew` – Amount of skew to be applied to indicators (default = 0). Data are first generated as random normal deviates (i.e., z scores), and then each score is skewed by raising e to the power of the score divided by `Skew`; in other words, $\text{skew}(x) = e^{(x/\text{Skew})}$. Using this method of skewing data, greater indicator skew is achieved by supplying *smaller* values for `Skew`. For example, a value of 1 generates substantial positive skew, whereas a value of 2 generates more moderate positive skew and a value of 4 generates comparatively mild positive skew. You can generate negatively skewed indicators by supplying negative values of `Skew`; in this case, the skewing function uses the absolute value of `Skew` to transform the scores and then subtracts them from 0 to create negatively skewed distributions. (When generating categorical data, the program preserves the desired taxon base rate despite the reversal of indicator scores; curious readers can inspect the program code or artificial categorical data sets to verify this.) The best way to get a feel for how different values of `Skew` influence the indicator distributions is to create data sets and examine the results. For example, if you create a data set called `Skew.Test` by typing `Skew.Test <- CreateData("Dim", Skew = 2)`, you can examine the first indicator's distribution using the command `plot(density(Skew.Test[,1]))` (replace the 1 with another number to plot a different indicator's distribution). Or, you can calculate skew on the conventional metric (where 0 represents no skew) by typing `Skew(Skew.Test[,1])`; see Chapter 6 for more information on the `Skew` function included in the suite of taxometric programs. Note that (1) introducing skew alters the correlations among indicators (usually, they are attenuated relative to what they would be for normally distributed data) and (2) data are restandardized after being skewed.

`Cuts` – Number of values to use when generating ordered categorical data (default = 0). If left at the default, data are not cut into ordered categories. If specified, the procedure first generates continuous data and then applies equal-spaced cutting scores along the distribution of continuous scores to yield `Cuts` categories on the final indicator (before locating cutting scores, the program trims 1%

of cases from each extreme of the distribution to reduce the influence of outliers). This technique is performed separately for each indicator, so their final distributions will vary from one another.

Seed – Random number seed provided prior to generating any data (default = 1). Specifying the same seed enables users to generate and analyze identical data sets.

3.5 Running the Program:

The following command line creates a categorical data set with 1000 cases, 4 indicators, a taxon base rate of .25, and validity of 2.00 SD:

```
TestCat <- CreateData(Str = "Cat", N = 1000, k = 4, P = .25, d = 2.00)
```

Similarly, the following command creates a dimensional data set with 1000 cases, 4 indicators, and the correlational equivalent of $P = .25$ and $d = 2.00$:

```
TestDim <- CreateData(Str = "Dim", N = 1000, k = 4, P = .25, d = 2.00)
```

Alternately, you could use the formula in Meehl and Yonce (1994, p. 1146) to determine the expected correlation (.43 in this case) and substitute $r = .43$ for specifications of the base rate and indicator validity when creating *TestDim*:

```
TestDim <- CreateData(Str = "Dim", N = 1000, k = 4, r = .43)
```

Also, because these commands happen to request some default values (the number of indicators and, for the categorical data, the indicator validity), the commands could be simplified:

```
TestCat <- CreateData(Str = "Cat", N = 1000, P = .25)
TestDim <- CreateData(Str = "Dim", N = 1000, r = .43)
```

As another example, the following command line creates a dimensional data set with 600 cases, 4 indicators that are moderately positively skewed and cut into 6 ordered categories, and $\text{Corr} = .43$:

```
SkewedDim <- CreateData(Str = "Dim", r = .43, Skew = 2, Cuts = 6)
```

Positively skewed indicators of a latent dimension, such as those in *SkewedDim*, are useful for examining how various taxometric procedures and consistency tests perform, as the results can often be mistaken for evidence of a small taxon. For example, a MAXEIG analysis of *SkewedDim* yields curves that appear peaked toward the right and consistently low estimates of the taxon base rate—which could readily lead one to a (mistaken) conclusion of categorical structure (see the sample commands and output for MAXEIG in Chapter 8). It is instructive to consider aspects of the results that lead to a (correct) conclusion of dimensional structure, such as a visual inspection of results for comparison data and a curve fit index that can be calculated based on these (for details, see the description of *GenData* in Chapter 4). Additionally, one could implement the inchworm consistency test by systematically increasing the number of windows in subsequent MAXEIG analyses—for categorical data, the base rate estimates should remain consistent across these analyses and a peak should become more clearly defined, whereas for dimensional data, the base rate estimates are likely to drop with additional windows and the curves should continue to rise without defining a clear peak (see J. Ruscio, Ruscio, & Keane, 2004).

3.6 Output:

This program produces no text output; it returns a data object. Therefore, the results of this program should be assigned to a new object for storage in the workspace. That is, rather than just calling the program, you name an object to hold the data that is generated and use the R assignment operator “<-” to assign the results of the command to the specified object (e.g., `TestCat <- CreateData(...)`). This usage contrasts with some of the taxometric programs that appear below, which are simply called to produce graphs and text output and are not assigned to objects. If you were to run the *CreateData* program without assigning its results to an object, the data that are generated would be displayed in the console window but unavailable for subsequent analysis.

Chapter 4: *GenData*

Generating Comparison Data Sets

4.1 Overview:

This program creates a finite population of comparison data that reproduces the indicator distributions and correlations observed in a data set supplied by the user. This is not a program that you should use directly; rather, it is called by the MAMBAC, MAXEIG, and L-Mode programs if you request the use of comparison data. This technique can be used (1) to help determine whether the research data are likely to yield informative taxometric results, (2) to provide a benchmark for comparison when interpreting results for the research data, and (3) to perform a consistency test based on the fit of the curves obtained for the research data to those obtained for categorical and dimensional comparison data. This approach and the *GenData* program represent update of an earlier approach that used programs called *DimSample* and *TaxSample*, which in turn replaced the original programs called *SimDim* and *SimTax* (for readers familiar with these older programs, the revisions will be described in later sections).

4.2 Using Comparison Data in Taxometric Studies:

The idiosyncratic distributional properties of real data (e.g., non-normal distributions, ordered categorical rather than continuous response scales) can have pronounced effects on taxometric results, and analyses of comparison data can help to reveal whether a curve shape is unique to a given latent structure or the result of some other aspect of the data. Likewise, one can determine whether the planned analyses are capable of distinguishing categorical from dimensional structure while holding constant the unique distributional and correlational characteristics of the research data. This approach has been described in a number of publications, including Ruscio (2007); Ruscio, Haslam, and Ruscio (2006); Ruscio and Kaczetow (2009); Ruscio and Marcus (2007); and Ruscio, Ruscio, and Meron (2007).

An overview of the approach is presented here, but the articles by Ruscio and Kaczetow (2009) and Ruscio, Ruscio, and Meron (2007) describe this in greater detail and provide extensive evidence of the performance of the programs to generate comparison data and the utility of analyzing these data to help interpret taxometric results. Also of note is that a paper by Beach, Amir, and Bau (2005) criticized the use of comparison data and programs for generating them. A reanalysis of Beach et al.'s data by Ruscio and Marcus (2007) corrects a number of factual errors and conceptual mistakes and then presents evidence that supports the use of comparison data in taxometric research.

GenData uses an iterative procedure to reproduce the observed indicator correlation matrix through loadings on one or more latent factors, and reproduces each indicator's distribution (including *M*, *SD*, skew, kurtosis, and ordered categorical values if applicable) through a standard bootstrap technique. Specifically, each indicator's observed distribution is used as an unbiased estimate of the population distribution, and values for the simulated indicator are resampled (with replacement) from this estimated population distribution (discussions of the bootstrap technique are available in many sources, e.g., Efron and Tibshirani, 1993). When the MAMBAC, MAXEIG, MAXSLOPE, or L-Mode program calls *GenData*, it does so once to create a finite population of dimensional comparison data, and twice to create finite populations of taxon and complement comparison data; these latter data are merged to yield a single population of categorical comparison data. From each of the two resulting populations, multiple samples of comparison data are drawn randomly for analysis.

Perhaps the most important aspect of using comparison data is to provide the most appropriate classification of cases when generating the categorical comparison data. *GenData* will reproduce indicator correlations and distributions well within groups *that you define*. In other words, if your assignment of cases to groups is highly fallible (e.g., many taxon and complement members are misclassified), the categorical comparison data will reflect this. For this reason, I recommend that careful attention be paid to the classification of cases, and I encourage a full description of the technique that was used in any research report including comparison data. Ruscio (2009) found that a base-rate classification technique (i.e., ranking cases by indicator total scores and cutting to form taxon and complement groups at a location determined by an estimated taxon base rate) achieved greater accuracy than the use of Bayes' Theorem. Both Ruscio and Kaczetow (2009) and Ruscio, Ruscio, and Meron (2007) found that using this method to provide the criterion for generating categorical comparison data worked well. An automated technique such as base-rate classification requires no knowledge about the construct under study, as does the use of Bayes' Theorem to assign cases to groups using parameters estimated through a taxometric analysis (e.g., MAXEIG or MAXCOV). However, researchers who can provide a more appropriate classification should do so. It should be noted, however, that even the lower-bound estimate of how accurately structure was identified using the base-rate classification technique was significantly higher than the accuracy with which any of several traditional taxometric consistency tests performed. When assessing the upper-bound accuracy of this approach by providing actual group membership to generate categorical comparison data, categorical structure was correctly identified in more than 99% of all cases, much better than any competing test.

Whereas the generation of categorical comparison data requires the thoughtful choice of a technique for assigning cases to groups, no classification of cases is required to generate dimensional comparison data.

Provided that categorical comparison data are generated in an appropriate way, comparison data should prove useful for several purposes. You can begin by generating multiple samples of dimensional and categorical comparison data and submitting each to the full array of planned taxometric analyses and consistency tests. If the obtained sampling distributions of results (graphs and statistics) for the categorical and dimensional comparison data can be distinguished from one another, this suggests that your research data will afford an informative test between these two structures.

If the sampling distributions of results for the dimensional and categorical comparison data cannot be distinguished, this suggests that your data may not afford an informative test, and consequently you should consider working with the data to meet the requirements of taxometric analysis more adequately and/or alter the program specifications to implement the analysis in a way that better differentiates categorical and dimensional structure. It may be unwise to proceed with a taxometric analysis (or consistency test) that has not been shown to yield discernibly different results for categorical and dimensional comparison data, for the results are likely to be ambiguous or, worse, misleading. For example, if your indicators are of insufficient validity to detect a taxon, the results for the research data are likely to appear consistent with dimensional structure and you might reach an incorrect structural inference on that basis. If, instead, you had performed parallel analyses of categorical and dimensional comparison data, you would have seen that the results appear consistent with dimensional structure even for a data set known to possess categorical structure. In addition to potentially fdimensional results, one might obtain false-categorical results. For example, positively skewed indicators of a latent dimension can produce taxometric curves with a right-end cusp that can be mistaken for a categorical peak. Thus, it may be safest to withhold judgment about structure when the results for categorical and dimensional comparison data cannot be differentiated, as otherwise you may increase the odds of reaching a mistaken conclusion.

The technique described above essentially evaluates the adequacy of the data for a planned taxometric analysis. It may well be the case that data are adequate for some, but not all, of the planned analyses (e.g., MAXEIG may distinguish the categorical and dimensional comparison data even if MAMBAC does not, or vice versa), in which case only the results for procedure(s) that pass the test should be used to draw conclusions about structure. In addition to empirically evaluating the adequacy of a particular data set for a particular analysis, another advantage of this approach is that you can try out a large number of analytic variations to determine how best to distinguish categorical and dimensional structure. The fact that categorical and dimensional comparison data produce distinct results demonstrates that the planned analysis may yield results consistent with either latent structure when performed on research data that share the same distributional and correlational characteristics of the research data.

4.3 An Objective Index of Curve Fit

If you believe that the data are adequate for one or more analyses and you have analyzed the research data in the appropriate way(s), the interpretation of results should be simplified by making reference to the output for the comparison data. Ideally, the results for either the categorical or the dimensional comparison data would more closely match those for your research data. To assist in the interpretive process, an index of curve fit is provided by the MAMBAC, MAXEIG, and L-Mode programs. Monte Carlo results suggest that this index performs better than several more traditional taxometric consistency tests (Ruscio, 2007; Ruscio, Ruscio, & Meron, 2007) and that it is robust across a wide range of data conditions, such as small taxa (Ruscio & Kaczetow, 2009; Ruscio & Marcus, 2007).

The first step in calculating this comparison curve fit index (CCFI) is to calculate Fit_{RMSR} , the root mean square residual (RMSR) of the values on the averaged curve for the research data and the averaged curve for either categorical or dimensional comparison data. For MAMBAC, MAXCOV, MAXEIG, or MAXSLOPE analyses, this is done using y (ordinate) values only (Eq. 1, below); for L-Mode analyses, this is done using the smallest Euclidean distance (Eq. 2), which incorporates both x and y values in the calculation of distance (Eq. 3). Prior to calculating the Euclidean distances for L-Mode analyses, (a) a single density plot is constructed using factor scores from all samples of comparison data drawn from each type of population (categorical and dimensional), (b) x values are rescaled to vary across a range of the same width as the y values, and (c) a parameter representing horizontal shift is optimized to minimize the Fit_{RMSR} value. See Ruscio and Walters (2009) for details on how the CCFI was adapted for use in L-Mode analyses and a simulation study that supports its utility.

$$(1) \quad Fit_{RMSR} = \sqrt{\frac{\sum (y_R - y_C)^2}{N}} \quad (2) \quad Fit_{RMSR} = \sqrt{\frac{\sum \min(dist)^2}{N}} \quad (3) \quad dist = \sqrt{(x_R - x_C)^2 + (y_R - y_C)^2}$$

Geometrically, for MAMBAC, MAXCOV, MAXEIG, and MAXSLOPE analyses the residuals are strictly vertical distances (y values only), whereas for L-Mode analyses the residuals are the shortest straight-line distances between curves. Calculating Fit_{RMSR} yields two values (one per structure) to be compared to one another. Perfect fit yields a value of 0, and poorer fit yields higher values, so the structure yielding a *lower* value of Fit_{RMSR} is better supported by the data. The CCFI integrates the two fit values into a single index

with theoretical anchors at 0 (strongest possible support for dimensional structure) and 1 (strongest possible support for categorical structure); .50 represents equivalent support for both structures. Specifically:

$$CCFI = Fit_{RMSR-dim} / (Fit_{RMSR-dim} + Fit_{RMSR-tax})$$

It is important to note that this is an index of the relative fit of these two structural models, not an index of either model's absolute goodness of fit. Two recent papers (Ruscio & Walters, 2009; Ruscio, Walters, Marcus, & Kaczetow, 2010) suggest that CCFI values between .40 and .60 are fairly ambiguous and should be interpreted with caution. For example, when the CCFI lies outside this range, there appears to be at least a 90% probability that it affords a valid structural differentiation; this probability is considerably higher as the CCFI diverges further from .50. On the other hand, when $.40 < CCFI < .60$, the probability of a valid structural differentiation is lower, much lower as the CCFI approaches .50, and it may be best to withhold judgment in these cases rather than risk an incorrect structural inference.

The MAMBAC, MAXEIG, MAXSLOPE, and L-Mode programs can be run in a way that seamlessly integrates the analysis of comparison data, in which case graphs are presented together for the research data, categorical comparison data, and dimensional comparison data (see examples in Chapters 7 – 10). The programs calculate the CCFI when comparison data are included in the analysis.

4.4 Differences Between DimSample and SimDim:

As noted earlier, DimSample is a replacement for the SimDim program. TaxSample is, analogously, a replacement for SimTax, but because both programs to generate categorical comparison data do nothing more than call the program to generate dimensional comparison data twice (once to reproduce the data within each group), they will not be discussed further. The revisions to DimSample apply to TaxSample as well. There are three important differences between DimSample and SimDim:

1. Whereas SimDim reproduced each indicator's distribution literally by copying it, DimSample does so by bootstrapping a distribution (as described earlier). This incorporates normal sampling error into each indicator's distribution.
2. Whereas SimDim reproduced indicator correlations using loadings onto a single latent factor, DimSample uses loadings onto one or more latent factors. The number of factors is determined through a factor analysis of the data, using the liberal criterion of counting the number of factors with an eigenvalue > 1 . This allows for multidimensionality in both dimensional and (within groups) categorical comparison data rather than constraining comparison data to unidimensionality.
3. DimSample reproduces indicator correlations using a more efficient iterative algorithm than SimDim. Interested readers can examine the program code to see how this is done. In tests using 10,000 target data sets with varying distributional and correlational characteristics, DimSample converged on its solution much more rapidly than SimDim, with no detectable loss in the precision with which correlations were reproduced (Ruscio, Ruscio, & Meron, 2007); there is no effect on distributions, as they are reproduced solely through the bootstrap.

4.5 Further Modifications of DimSample:

The DimSample program has been modified since the version that was published in the Appendix to Ruscio, Ruscio, and Meron (2007). Specifically, two changes have been made:

1. The program used to contain a loop over all cases in the data set nested within a loop over all indicators. The former loop was removed, which improves the speed of the program, especially with large samples. This change has no effect on the output of the program, it simply accelerates processing.
2. The criterion to determine the number of factors for reproducing the indicator correlation matrix was changed. Previously, the program used the liberal Kaiser criterion of the number of eigenvalues > 1 . Now, a parallel analysis is performed using 100 random data sets (see Reise, Waller, & Comrey, 2000).

4.6 GenData Replaces DimSample and TaxSample:

The GenData program developed by Ruscio and Kaczetow (2008), and applied to taxometrics by Ruscio and Kaczetow (2009), is extremely similar to the DimSample program. Its use in the updated taxometric programs obviates the need for a separate program to generate categorical comparison data (the TaxSample program). As described earlier, the taxometric programs call GenData as needed to reproduce data of both structures.

Chapter 5: *Indicator.Dist*

Describing Indicator Distributions, Correlations, and Validity for a Data Set

5.1 Overview:

This program calculates summary statistics to describe indicator distributions (in the full sample and within groups), the indicator correlation matrix (also in the full sample and within groups), and the validity with which each indicator distinguishes groups. The program requires the user to include a variable containing the group membership (taxon or complement) of each case. In the event that one is uninterested in within-group statistics or between-group validity, one can ignore those pieces of the output but the program still requires that all cases be assigned to groups and that neither group be empty.

5.2 Command:

```
Indicator.Dist <- function(Data)
```

5.3 Required Arguments:

Data – The target data set, which must include a final column containing group membership, coded as 1 = complement, 2 = taxon

5.4 Running the Program:

The following command line examines the `TestCat` data set (previously generated using `CreateData`):

```
Indicator.Dist(TestCat)
```

5.5 Output:

This program produces many tables of output. First, the program prints the *M*, *SD*, skew, and kurtosis of each indicator, along with the *M* and *SD* of each of these summary statistics across indicators. This is done in the full sample and then within each group (taxon and complement). Second, the program prints validity for each indicator (calculated in both “raw” and standardized units, with the latter being Cohen’s *d*), and a table summarizes these results. Third, the program prints the indicator correlation matrices in the full sample, the taxon, and the complement, followed by a summary of the correlations in these matrices. Sample output appears on the next page.

> Indicator.Dist(TestCat)

Indicator distributions in the full sample (N = 1000):

	M	SD	Skew	Kurtosis
Ind 1	0	1	0.298	-0.053
Ind 2	0	1	0.407	0.037
Ind 3	0	1	0.293	-0.266
Ind 4	0	1	0.307	-0.208
M	0	1	0.326	-0.122
SD	0	0	0.054	0.139

Indicator distributions in the taxon (n = 250):

	M	SD	Skew	Kurtosis
Ind 1	1.130	0.762	-0.005	0.124
Ind 2	1.175	0.767	0.002	0.322
Ind 3	1.123	0.733	-0.119	0.112
Ind 4	1.128	0.758	-0.225	0.827
M	1.139	0.755	-0.087	0.346
SD	0.024	0.015	0.108	0.335

Indicator distributions in the complement (n = 750):

	M	SD	Skew	Kurtosis
Ind 1	-0.377	0.757	-0.089	0.023
Ind 2	-0.392	0.724	-0.036	0.216
Ind 3	-0.374	0.771	0.113	0.098
Ind 4	-0.376	0.759	0.041	-0.087
M	-0.380	0.753	0.007	0.062
SD	0.008	0.020	0.088	0.127

Between-group validity:

	Raw Units	Cohen's d
Ind 1	1.506	1.986
Ind 2	1.567	2.133
Ind 3	1.497	1.966
Ind 4	1.504	1.982
M	1.519	2.017
SD	0.032	0.078

Indicator correlations:

Full Sample (N = 1000):

	Ind 1	Ind 2	Ind 3	Ind 4
Ind 1	1.000	0.433	0.440	0.452
Ind 2	0.433	1.000	0.446	0.415
Ind 3	0.440	0.446	1.000	0.433
Ind 4	0.452	0.415	0.433	1.000

Taxon (n = 250):

	Ind 1	Ind 2	Ind 3	Ind 4
Ind 1	1.000	0.021	-0.004	0.020
Ind 2	0.021	1.000	0.068	-0.117
Ind 3	-0.004	0.068	1.000	-0.115
Ind 4	0.020	-0.117	-0.115	1.000

Complement (n = 750):

	Ind 1	Ind 2	Ind 3	Ind 4
Ind 1	1.000	-0.032	0.039	0.056
Ind 2	-0.032	1.000	-0.010	-0.024
Ind 3	0.039	-0.010	1.000	0.061
Ind 4	0.056	-0.024	0.061	1.000

Summary of indicator correlations:

	M	SD
Full Sample	0.437	0.013
Taxon	-0.021	0.077
Complement	0.015	0.042

Chapter 6: Skew & Kurtosis

Calculating the Skew and Kurtosis of a Distribution

6.1 Overview:

These functions calculate the familiar descriptive statistics, which are not otherwise available in R. The formulas used for each are as follows:

$$m_i = \frac{\sum (X - \bar{X})^i}{N}; \quad g_1 = \frac{m_3}{m_2^{3/2}}; \quad g_2 = \frac{m_4}{m_2^2} - 3$$
$$skew = \left(\frac{\sqrt{N(N-1)}}{N-2} \right) g_1; \quad kurtosis = \left(\frac{N-1}{(N-2)(N-3)} \right) ([N+1]g_2 + 6)$$

6.2 Commands:

```
Skew <- function(x)
Kurtosis <- function(x)
```

6.3 Required Arguments:

`x` – The supplied variable (*not* an entire data set with multiple variables) whose skewness or kurtosis is to be calculated

6.4 Running the Program:

The following command lines would calculate the skewness and kurtosis for the first indicator in the `SkewedDim` data set (previously generated using `CreateData`; see Chapter 3):

```
Skew(SkewedDim[,1])
Kurtosis(SkewedDim[,1])
```

6.5 Output:

Each function reports a single value as output. For example, the skewness and kurtosis for the first indicator in `SkewedDim` are:

```
> Skew(SkewedDim[,1])
[1] 0.8952685

> Kurtosis(SkewedDim[,1])
[1] 0.4305128
```

Chapter 7: MAMBAC

Performing Mean Above Minus Below A Cut (MAMBAC) Taxometric Analyses

7.1 Overview:

This program runs the mean above minus below a cut (MAMBAC) taxometric procedure introduced by Meehl and Yonce (1994).

7.2 Command:

```
MAMBAC <- function(Data.Set, Comp.Data = T, N.Pop = 100000, N.Samples = 100, Supplied.Class = F, Supplied.P = 0,
  All.Pairs = T, Ind.Comp = F, N.End = 25, N.Cuts = 50, St.Ind = F, Replications = 0,
  Gr.Comp.Only = T, Gr.Rows = 3, Gr.Cols = 2, Gr.Smooth = F, Gr.Common.Y = F, Gr.Cases = T,
  Gr.Avg = F, Gr.Ind = F, Gr.Base.Rates = F, File.Output = F, File.Name = "Output.txt",
  Seed = 1)
```

7.3 Required Arguments:

Data.Set – The supplied data file, which must contain one case per row, one indicator per column. If a variable is included to signify group membership for each case (1 = complement, 2 = taxon), this must be in the final column of the data provided to the program (see **Supplied.Class**, below). Cases missing any data will be removed prior to analysis.

7.4 Optional Arguments:

Comp.Data – Whether to generate and analyze categorical and dimensional comparison data (default = T). When this is set to T, comparison data are used (see Chapter 4 for details) and, because this involves the averaging of curves, indicators are standardized.

N.Pop – The size of the finite populations of categorical and dimensional comparison data (default = 100,000). Unless the number of indicators is unusually large, this should run reasonably quickly.

N.Samples – Number of comparison data sets of each structure to generate and analyze (default = 100). Generating multiple sets of comparison data is strongly encouraged, as it allows one to examine a sampling distribution of results for each structure.

Supplied.Class – Whether the final column of the supplied data set contains group membership (coded as 1 = complement, 2 = taxon) to use for the estimation of data parameters and the generation of categorical comparison data (default = F).

Supplied.P – As an alternative to providing group membership, the program will accept a user-specified taxon base rate and assign cases to groups using the base-rate classification method described in Chapter 4 (default = 0). If this value is left at 0 *and* group membership is not provided, the program assigns cases to groups using the estimated base rate.

All.Pairs – Whether to use variables in all input-output pairings (default = T).

Ind.Comp – Whether to use composite input indicators by summing all but the output indicator to form each input indicator (default = F). Note that setting this to T overrides **All.Pairs = T**.

N.End – The number of cases to set aside at each extreme along the input indicator before making the first and last cut (default = 25). This protects the stability of the curve at its extremes.

N.Cuts – Total number of cuts to make along the input indicator (default = 50).

St.Ind – Whether to standardize each indicator prior to analysis (default = F). As noted above, if comparison data are used indicators are standardized because curves are averaged for presentation.

Replications – Number of times to resort cases along the input indicator at random and redo the calculations, averaging to obtain final results (default = 1). Replications are only of use when the locations of cuts are often arbitrarily placed between equal-scoring cases, in which event the replication procedure minimizes the sampling error that arises from such arbitrary ordering of tied cases. The program will check for tied scores, and if any are found **Replications** will be set to 10 (unless another value was specified); if none are found, **Replications** will be set to 1 regardless of what was specified because additional analyses would not change results.

Gr.Comp.Only – Whether to restrict graphical output to the 2-panel plot that includes results for comparison data (default = T).

Gr.Rows – Number of rows of graphs on a page (default = 3).

`Gr.Cols` – Number of columns of graphs on a page (default = 2).

`Gr.Smooth` – Whether to add a smoothed line (using the LOWESS method) to each graph (default = F). Note that if this option is chosen, smoothed values are also used in the estimation of the taxon base rate. This can be useful if the curve is unstable at its extremes, as the base rate is estimated using only the two endpoints of the curve.

`Gr.Common.Y` – Whether to apply the same Y scale to all graphs obtained through the analysis of a given data set (default = F).

`Gr.Cases` – Whether to scale and label each x axis by case numbers (default = T). If this option is set to F, the x axis will be scaled and labeled according to scores on input indicators. The latter option is only advisable when the input indicators take on a large number of distinct values, otherwise the points will be arranged oddly and the curve may be less interpretable.

`Gr.Avg` – Whether to plot a single, averaged curve (default = F). If this option is chosen, an averaged curve will be plotted (as a solid line amidst dotted lines representing each individual curve) in addition to the full panel of curves and, if selected, curves averaged for each input indicator (see `Gr.Ind`, below). Note that if comparison data are used, the program will automatically work with an averaged curve and indicators are standardized for this reason. Also, curves can be averaged only if MAMBAC values are calculated by case numbers, not by input indicator scores (see `Gr.Cases`, above).

`Gr.Ind` – Whether to plot an averaged curve for each input indicator (default = F). If this option is chosen, an averaged curve will be plotted for each input indicator (as a solid line amidst dotted lines representing each individual curve for that indicator) in addition to the full panel of curves and, if selected, a single averaged curve (see `Gr.Avg`, above). Also, curves can be averaged only if MAMBAC values are calculated by case numbers, not by input indicator scores (see `Gr.Cases`, above).

`Gr.Base.Rates` – Whether to summarize and plot the sampling distributions of base rate estimates for comparison data (default = F). If this option is chosen, information will appear in the text output as well as in a graph window.

`File.Output` – Whether to send text output to a file rather than displaying it on the screen (default = F).

`File.Name` – The filename to use when text output is sent to a file (default = “Output.txt”). Note that (1) this has no effect if `File.Output = F` and (2) this file appears in your default R directory unless you specify a full path along with the filename.

`Seed` – Random number seed provided prior to analysis of empirical data as well as prior to generating each population of comparison data (if comparison data are used); the default value is 1. In addition to affording exact replications of analyses, this enables the user to generate and analyze identical populations and samples of comparison data across analyses using different taxometric procedures. To ensure that identical populations and samples are used, make sure that `N.Pop` and `N.Samples` are held constant across analyses and that the same classification of cases is used to generate categorical comparison data. This latter requirement can be achieved by providing classification codes (and setting `Supplied.Class = T`) or by providing the same taxon base rate estimate for base-rate classification (setting `Supplied.P` to the same value for each analysis).

7.5 Running the Program:

The program automatically determines your sample size and the number of indicators. Take care to include only indicator variables, one per column. If you are supplying an additional column with group membership for generating categorical comparison data, make sure that you specify `Supplied.Class = T`; otherwise, this column is read as an indicator variable.

The following commands would conduct a MAMBAC analysis of `TestCat` using without comparison data (but otherwise using all program defaults) and then re-run the analysis using comparison data:

```
MAMBAC(TestCat[,1:4], Comp.Data = F)
MAMBAC(TestCat[,1:4])
```

There are many parameters that can be varied for convenience (e.g., the number of rows and columns on graph pages), but one particular issue should be carefully considered when conducting MAMBAC. By default, the MAMBAC program uses all variables in the data file in all possible input-output pairings, going in both directions (e.g., three variables X, Y, and Z would result in the six pairings X-Y, Y-X, X-Z, Z-X, Y-Z, and Z-Y—this occurs when `All.Pairs = T`). This is the traditional way to perform MAMBAC. An alternative approach is to use each variable in the data file once as an output indicator, with the remainder of the variables combined to form the input indicator. The MAMBAC program uses composite input indicators when you set `Ind.Comp = T`. Enthusiasm for the use of composite input indicators has been expressed in many sources, but a study by Walters and Ruscio (2009) found that this technique for accommodating data with ordered categorical response scales provided little or no advantage relative to the traditional way to perform MAMBAC (or MAXCOV/MAXEIG). It remains possible that composite input indicators may improve the power of some taxometric procedures under certain conditions (e.g., when all indicators are valid), but this has not been tested. The choice of how many cases to reserve beyond the first and last cuts was studied recently (Walters & Ruscio, 2010), with conditions of $n = 10, 25, 50$, 5% of total N , and 10% of total N . Results did not differ appreciably across these five implementations, nor were

they sensitive to the taxon base rate (which varied from .10 to .50 in this study). In light of these results, the program default was left at $n = 25$.

7.6 Output:

The program plots a panel of MAMBAC curves (unless output is restricted to the 2-panel plot including results for comparison data). If requested, one can also obtain a panel of MAMBAC curves averaged for each input indicator, and/or a single averaged curve. Each curve in a panel consists of mean differences on the output indicator along the y axis, with cutting scores on the input indicator along the x axis. Each graph is fully labeled, clearly communicating which indicator variable(s) served as input and output and whether cases or indicator scores are used to scale and label the x axis. If comparison data are analyzed, a 2-panel graph sheet presents the results for the research data (dark line with points) superimposed above the results for the categorical comparison data, and then the results for the dimensional comparison data; results for comparison data sets are summarized by plotting the middle 50% of data points as a gray band and light lines that show the minimum and maximum values.

In addition, a detailed text summary of the analysis is provided. This summary includes an overview of relevant data parameters and program specifications, base rate estimates calculated from each curve (listed in the order in which the curves are plotted) with a summary (M , SD) of estimates across curves, and—in cases in which each indicator served as input for more than one curve—base rate estimates collapsed for each indicator variable with a summary (M , SD) of estimates across indicators. Cases are classified using the mean base rate estimate: Those with the highest total scores on all indicators are assigned to the taxon, the remainder to the complement, such that the size of the taxon will equal the estimated base rate. Using this classification, all output of the `Indicator.Dist` program is provided (see Chapter 6). If comparison data are generated and analyzed, a summary of the accuracy with which correlations were reproduced for each data set is provided along with the M s and SD s of the base rate estimates for each sample of comparison data and a curve fit index (CCFI; see Chapter 4). Sample output follows. Throughout this manual, accompanying graphs immediately follow the associated text output for a given command.

```
> MAMBAC(TestCat[,1:4])
```

SUMMARY OF MAMBAC ANALYTIC SPECIFICATIONS

```
Sample size: 1000
Number of indicator variables: 4
Replications: 1
Cuts: 50 evenly-spaced cuts beginning 25 cases from either extreme
Indicators: Variables serve in all possible input-output pairs
Total number of curves: 12
Y values smoothed for graphing and P estimation: No
Classification of cases: Cut total score at estimated base rate
Size of finite populations of comparison data: 1e+05
Number of samples of comparison data drawn from each population: 100
```

SUMMARY OF MAMBAC PARAMETER ESTIMATES

Estimated taxon base rates for each curve:

	P
Curve 1	0.140
Curve 2	0.117
Curve 3	0.277
Curve 4	0.303
Curve 5	0.313
Curve 6	0.206
Curve 7	0.134
Curve 8	0.278
Curve 9	0.383
Curve 10	0.261
Curve 11	0.403
Curve 12	0.207

```
Summary of base rate estimates across curves:
  M = 0.252
  SD = 0.094
```

[continued on the next page]

Estimated taxon base rates for each indicator:

P
Indicator 1 0.243
Indicator 2 0.212
Indicator 3 0.328
Indicator 4 0.225

Summary of base rate estimates across indicators:

M = 0.252
SD = 0.052

Base rate estimate for averaged curve = 0.252

Indicator distributions in the full sample (N = 1000):

	M	SD	Skew	Kurtosis
Ind 1	0	1	0.298	-0.053
Ind 2	0	1	0.407	0.037
Ind 3	0	1	0.293	-0.266
Ind 4	0	1	0.307	-0.208
M	0	1	0.326	-0.122
SD	0	0	0.054	0.139

Indicator distributions in the taxon (n = 252):

	M	SD	Skew	Kurtosis
Ind 1	1.130	0.766	-0.057	0.166
Ind 2	1.169	0.754	0.196	-0.202
Ind 3	1.153	0.703	0.011	-0.335
Ind 4	1.128	0.742	-0.083	0.582
M	1.145	0.741	0.017	0.053
SD	0.020	0.027	0.126	0.412

Indicator distributions in the complement (n = 748):

	M	SD	Skew	Kurtosis
Ind 1	-0.381	0.751	-0.117	0.019
Ind 2	-0.394	0.728	-0.014	0.212
Ind 3	-0.389	0.756	0.052	0.064
Ind 4	-0.380	0.761	0.058	-0.082
M	-0.386	0.749	-0.005	0.053
SD	0.007	0.014	0.081	0.122

Between-group validity:

	Raw Units	Cohen's d
Ind 1	1.510	2.000
Ind 2	1.563	2.127
Ind 3	1.542	2.075
Ind 4	1.508	1.994
M	1.531	2.049
SD	0.027	0.064

Indicator correlations:

Full Sample (N = 1000):

	Ind 1	Ind 2	Ind 3	Ind 4
Ind 1	1.000	0.433	0.440	0.452
Ind 2	0.433	1.000	0.446	0.415
Ind 3	0.440	0.446	1.000	0.433
Ind 4	0.452	0.415	0.433	1.000

Taxon (n = 252):

	Ind 1	Ind 2	Ind 3	Ind 4
Ind 1	1.000	0.032	-0.044	-0.009
Ind 2	0.032	1.000	0.009	-0.090
Ind 3	-0.044	0.009	1.000	-0.149
Ind 4	-0.009	-0.090	-0.149	1.000

[continued on the next page]

Complement (n = 748):

	Ind 1	Ind 2	Ind 3	Ind 4
Ind 1	1.000	-0.042	0.015	0.056
Ind 2	-0.042	1.000	-0.025	-0.040
Ind 3	0.015	-0.025	1.000	0.033
Ind 4	0.056	-0.040	0.033	1.000

Summary of indicator correlations:

	M	SD
Full Sample	0.437	0.013
Taxon	-0.042	0.068
Complement	0.000	0.041

Generating and analyzing comparison data

Categorical population generated, RMSR r = 0.005

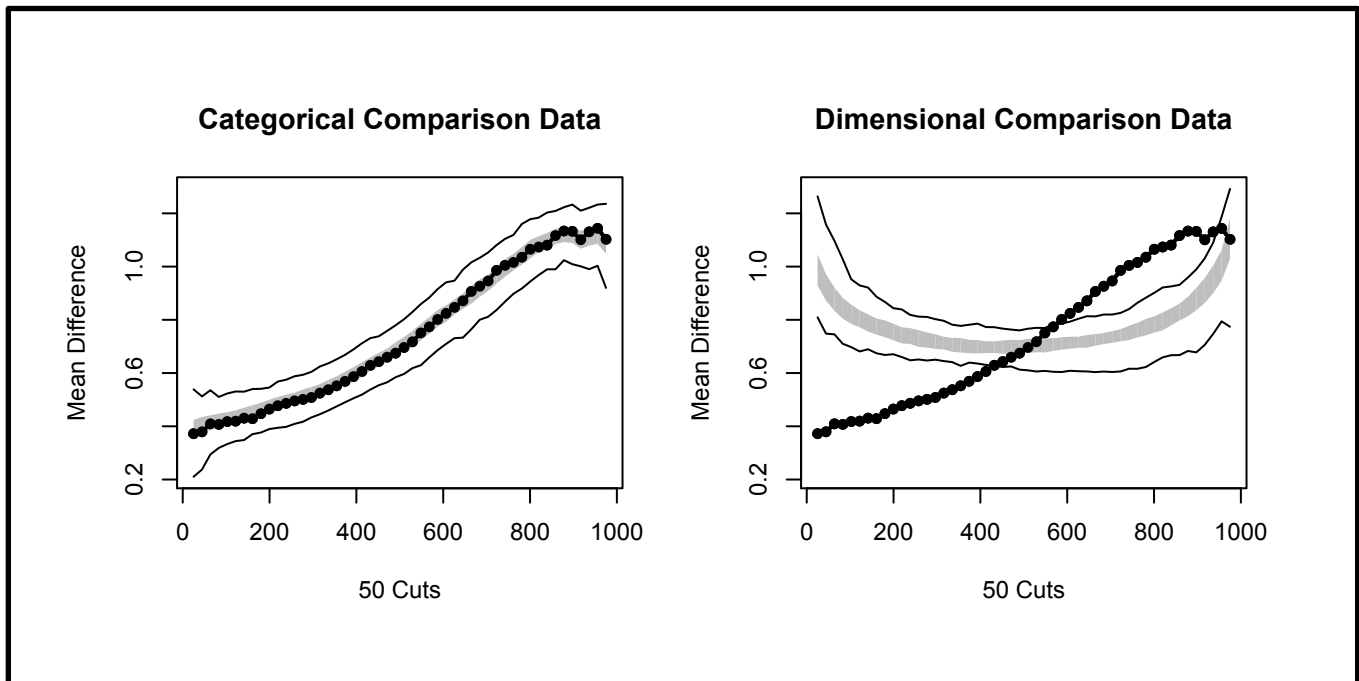
Analysis of 100 samples of categorical comparison data completed

Dimensional population generated, RMSR r = 0.01

Analysis of 100 samples of dimensional comparison data completed

Comparison curve fit index (CCFI) = $0.2561 / (0.2561 + 0.015) = 0.945$

Note: CCFI values can range from 0 (dimensional) to 1 (categorical). The more a CCFI value deviates from .50, the stronger the result. When $.40 < \text{CCFI} < .60$, this should be interpreted with caution.



Chapter 8: MAXEIG

Performing Maximum Covariance (MAXCOV) or Maximum Eigenvalue (MAXEIG) Taxometric Analyses

8.1 Overview:

This program is used to run the maximum covariance (MAXCOV) or the maximum eigenvalue (MAXEIG) taxometric procedures introduced by Meehl and Yonce (1996) and Waller and Meehl (1998), respectively. As noted below, the crucial difference is whether the program is asked to calculate eigenvalues or covariances; otherwise, these procedures are similar to one another in most important ways and therefore performed using the same program. When only two indicators are available, the maximum slope (MAXSLOPE) procedure can be performed instead of MAXCOV or MAXEIG.

8.2 Command:

```
MAXEIG <- function(Data.Set, Comp.Data = T, N.Pop = 100000, N.Samples = 100, Supplied.Class = F, Supplied.P = 0,
  Ind.Triplets = F, Ind.Comp = F, Intervals = F, N.Int = 15, Windows = 25, Overlap = .90,
  Calc.Cov = F, St.Ind = F, Replications = 0, Gr.Comp.Only = T, Gr.Rows = 3, Gr.Cols = 2,
  Gr.Smooth = 0, Gr.Common.Y = F, Gr.Ref = 2, Gr.Avg = F, Classify = 1, Gr.Bayes = F,
  Gr.Base.Rates = F, Save.Class = 0, File.Output = F, File.Name = "Output.txt", Seed = 1)
```

8.3 Required Arguments:

Data.Set – The supplied data file, which must contain one case per row, one indicator per column. If a variable is included to signify group membership for each case (1 = complement, 2 = taxon), this must be in the final column of the data provided to the program (see **Supplied.Class**, below). Cases missing any data will be removed prior to analysis.

8.4 Optional Arguments:

Comp.Data – Whether to generate and analyze categorical and dimensional comparison data (default = T). When this is set to T, comparison data are used (see Chapter 4 for details) and, because this involves the averaging of curves, indicators are standardized.

N.Pop – The size of the finite populations of categorical and dimensional comparison data (default = 100,000). Unless the number of indicators is unusually large, this should run reasonably quickly.

N.Samples – Number of comparison data sets of each structure to generate and analyze (default = 100). Generating multiple sets of comparison data is strongly encouraged, as it allows one to examine a sampling distribution of results for each structure.

Supplied.Class – Whether the final column of the supplied data set contains group membership (coded as 1 = complement, 2 = taxon) to use for the estimation of data parameters and the generation of categorical comparison data (default = F).

Supplied.P – As an alternative to providing group membership, the program will accept a user-specified taxon base rate and assign cases to groups using the base-rate classification method described in Chapter 4 (default = 0). If this value is left at 0 *and* group membership is not provided, the program checks the value of **Classify** to determine how to assign cases to groups (see below).

Ind.Triplets – Whether to use variables in all input, output, output triplets (default = F).

Ind.Comp – Whether to use composite input indicators by summing all but the output indicators to form each input indicator (default = F). Note that setting this to T overrides **Ind.Triplets** = T. When both **Ind.Triplets** and **Ind.Comp** are set to F (as is the default), each variable will serve once as the input indicator, with all remaining variables serving as output indicators.

Intervals – Whether to use nonoverlapping input intervals (default = F). By default, overlapping windows will be used instead.

N.Int – Number of equal-sized intervals for subdividing the input indicator (default = 15). Note that either of the next two options can be used to override this method.

Windows – Number of overlapping windows to use if **Intervals** = F (default = 50).

Overlap – Amount of overlap between successive windows (default = .90).

Calc.Cov – Whether to calculate covariances within subsamples, as is done in MAXCOV analyses (default = F). The calculation of covariances requires the use of just two output indicators per curve, hence **Ind.Triplets** is automatically set to T if **Calc.Cov** = T (**Ind.Comp** may also be set to T, as this also involves just two output indicators). If this parameter is left at F, eigenvalues are

calculated instead. This is the critical program parameter that determines whether one is performing a MAXCOV or a MAXEIG analysis, and all output is labeled accordingly.

`St.Ind` – Whether to standardize each indicator prior to analysis (default = F). Note that indicators are automatically standardized when dividing the input into subsamples by *SD* units, but composite input indicators are not restandardized after summation. Also, as noted above, if comparison data are used indicators are standardized because curves are averaged for presentation.

`Replications` – Number of times to resort cases along the input indicator at random and redo the calculations, averaging to obtain final results (default = 1). Replications are only of use when subsamples are determined by cuts arbitrarily placed between equal-scoring cases, in which event the replication procedure minimizes the sampling error that arises from such arbitrary ordering of tied cases. The program will check for tied scores, and if any are found `Replications` will be set to 10 (unless another value was specified); if none are found, `Replications` will be set to 1 regardless of what was specified because additional analyses would not change results.

`Gr.Comp.Only` – Whether to restrict graphical output to the 2-panel plot that includes results for comparison data (default = T).

`Gr.Rows` – Number of rows of graphs on a page (default = 3).

`Gr.Cols` – Number of columns of graphs on a page (default = 2).

`Gr.Smooth` – Whether to add a smoothed line using either the LOWESS (1) or running medians (2) methods (default = 0). Setting this to 0 turns off curve smoothing. Note that if either smoothing method is used, smoothed values are also used in the estimation of latent parameters.

`Gr.Common.Y` – Whether to apply the same Y scale to all graphs obtained through the analysis of a given data set (default = F). Note that even if this option is chosen, the Y scale is recalculated for each comparison data set that is analyzed.

`Gr.Ref` – Whether to graph a reference line at $Y = 0$ (default = 2). Acceptable values are 1 (always), 2 (as needed, meaning only on plots that extend below 0), or 3 (never).

`Gr.Avg` – Whether to plot a single, averaged curve (default = F). If this option is chosen, an averaged curve will be plotted (as a solid line amidst dotted lines representing each individual curve) in addition to the full panel of curves and, if selected, curves averaged for each input indicator (see `Gr.Ind`, below). Note that if comparison data are used, the program will automatically work with an averaged curve and indicators are standardized for this reason. Also, curves can be averaged only if a fixed number of intervals or windows are used, rather than SD intervals.

`Gr.Ind` – Whether to plot an averaged curve for each input indicator (default = F). If this option is chosen, an averaged curve will be plotted for each input indicator (as a solid line amidst dotted lines representing each individual curve for that indicator) in addition to the full panel of curves and, if selected, a single averaged curve (see `Gr.Avg`, above). Also, curves can be averaged only if a fixed number of intervals or windows are used, rather than SD intervals.

`Classify` – Whenever the user does not supply a taxon base rate or a classification of cases, this parameter determines whether to classify cases using the base-rate method with the estimated base rate (1, the default value) or Bayes' Theorem (2). Note that Bayes' Theorem cannot be used with composite input indicators, as one cannot estimate valid and false positive rates for individual indicators. If this input indicator method is used, the program automatically sets `Classify` to 1.

`Gr.Bayes` – Whether to plot the distribution of Bayesian probabilities of taxon membership (default = F). Note that this can only be set to T if the Bayesian classification method was used (see `Classify`, above).

`Gr.Base.Rates` – Whether to summarize and plot the sampling distributions of base rate estimates for comparison data (default = F). If this option is chosen, information will appear in the text output as well as in a graph window.

`Save.Class` – Whether to return output regarding class membership (default = 0). Acceptable values are 1 (return class assignments, coded as 1 = complement, 2 = taxon) or 2 (return Bayesian probabilities of taxon membership). Note that this latter option is available only if the Bayesian classification method was used (see `Classify`, above).

`File.Output` – Whether to send text output to a file rather than displaying it on the screen (default = F).

`File.Name` – The filename to use when text output is sent to a file (default = "Output.txt"). Note that (1) this has no effect if `File.Output = F` and (2) this file appears in your default R directory unless you specify a full path along with the filename.

`Seed` – Random number seed provided prior to analysis of empirical data as well as prior to generating each population of comparison data (if comparison data are used); the default value is 1. In addition to affording exact replications of analyses, this enables the user to generate and analyze identical populations and samples of comparison data across analyses using different taxometric procedures. To ensure that identical populations and samples are used, make sure that `N.Pop` and `N.Samples` are held constant across analyses and that the same classification of cases is used to generate categorical comparison data. This latter requirement can be achieved by providing classification codes (and setting `Supplied.Class = T`) or by providing the same taxon base rate estimate for base-rate classification (setting `Supplied.P` to the same value for each analysis).

8.5 Running the Program:

Whereas MAXCOV and MAXEIG have traditionally been presented and conceived as distinct procedures, they can be considered variations on the same basic approach. In both cases, a measure of association (either covariance or eigenvalue) is calculated among two (or more) output indicators within ordered subsamples along an input indicator. The approach taken in writing the MAXEIG program was to blend and extend the procedures, taking advantage of features of both to produce an even more flexible and powerful combination. In part because of this synthesis, there are a number of important decisions to make in running the MAXEIG program. It is to be stressed at the outset that relying on the program defaults is *often* likely to be inappropriate for a particular investigation. The most crucial of these decisions are briefly discussed below.

First, you need to determine how to configure the available variables to form input and output indicators. This was discussed for MAMBAC (see Chapter 7), though there are more choices for MAXEIG. First, you can use all variables in every input, output, output triplet (this is the traditional method for MAXCOV; use `Ind.Triplets = T`, which is not the default setting). This yields the greatest number of curves (and base rate estimates), which is good for consistency testing, yet it does not include all variables in each analysis when there are more than three variables available. Second, you can remove two variables to serve as output indicators, and combine the remaining variables to form a composite input indicator (using `Ind.Comp = T`, which is also not the default setting). This yields fewer curves, but it does involve all variables in every analysis. Both of these methods can be done using covariances, as there are only two output indicators in each case. The third method, which involves the computation of eigenvalues, is to remove one variable to serve as the input indicator and allow all of the remaining indicators to serve as output indicators. Like the second method, this reduces the number of curves and includes all variables in each analysis.

Second, you need to decide how to divide cases into subsamples along the input indicator. Again, there are different methods available. You can use a fixed number of equal-sized subsamples that may overlap to any desired degree (for nonoverlapping intervals, use `Intervals = T` and set `N.Int` to the desired number of intervals [default = 15]; for overlapping windows, use `Intervals = F` [the program default] and choose the number of `Windows` [default = 50] and degree of `Overlap` [default = .90]). Overlapping windows essentially fill in points between those that would be obtained using nonoverlapping intervals, and therefore should provide more interpretable curves. With either intervals or windows, the divisions between subsamples can occur among equal-scoring cases, and if so an arbitrary ordering of these cases determines how they are allotted to subsamples. To reduce the impact of such arbitrary grouping, the program allows for internal replications of the analysis, resorting cases each time subsamples are constructed and averaging covariances or eigenvalues across all replications. Particularly when small samples or subsamples are used, setting `Replications` to higher values can improve the stability of the resulting curve by counteracting the influence of arbitrary groupings.

Setting the number of subsamples forces you to confront the tradeoff between increasing the number of points on the curve (by using more subsamples) and decreasing sampling error at each point (by using fewer subsamples). Smaller subsamples can also be helpful in detecting particularly small taxa, which can be buried within a subsample that consists primarily of complement members. Thus, one potentially useful approach is to use the inchworm consistency test (Waller and Meehl, 1998) by repeating an analysis several times with increasingly small subsamples. If a small taxon exists, a peak should become better defined toward the end of the curve, whereas if no such taxon exists, an apparent peak may be shown to result from sampling error and disappear (or, in the case of skewed indicators of a latent dimension, a cusped curve may remain cusped rather than becoming peaked; Ruscio, Ruscio, & Keane, 2004).

A recent study of MAXCOV and MAXEIG implementation (Walters & Ruscio, 2010) examined 24 variations, varying each of four factors in a fully crossed design: (1) MAXCOV (calculating covariances using all possible indicator triplets) vs. MAXEIG (calculating eigenvalues using one variable as input indicator, all others as output indicators), (2) overlapping windows vs. nonoverlapping intervals, (3) choosing the number of subsamples (and thereby determining n within each indirectly) vs. choosing n per subsample (and thereby determining the number of subsamples indirectly), and (4) including more vs. fewer cases per subsample (equivalently, using fewer vs. more subsamples). The results provided strong support for using windows rather than intervals, choosing the number of windows, and choosing a small number of windows. The greatest accuracy was observed using 25 windows, in which case it made almost no difference whether one used MAXCOV or MAXEIG; the latter achieved slightly greater accuracy. In light of these findings, the program default of MAXEIG with 50 windows was changed to MAXEIG with 25 windows.

Third, you need to decide how to graph the results. A look at published taxometric research shows tremendous variation. For example, axes are often poorly labeled, and graphs sometimes fail to communicate key analytic details (e.g., what variables served as input/output indicators). The program described here clearly labels all axes to reflect the methods used to allocate variables to the roles of input and outputs, to subdivide the input (e.g., intervals or windows), and to calculate y values (e.g., covariances or eigenvalues).

Another concern involves the scaling of the y axis. Too narrow a range may exaggerate normal sampling error, creating apparent peaks, whereas too wide a range may flatten even genuinely peaked curves. To help resolve this problem, the MAXEIG program contains an algorithm that is designed to accentuate the difference between peaked and flat curves. It works by examining the variation in y values relative to the distance from the smallest value to the x axis. Whereas a peaked curve typically exhibits

considerable variability across its span and extends downward close to zero at one or both ends, a flat curve typically exhibits little variability and remains well above zero at all points. The scaling algorithm takes advantage of this difference to choose a range of values that tends to draw out true peaks without transforming normal sampling error into apparent peaks. (Interested readers can consult the program code for details.) This effectiveness of this technique diminishes with increasing indicator skew, which can cause rising curves that also possess considerable variability across data points relative to the space beneath the lowest value.

Despite clear labeling and an analytic attempt to derive useful y scales, at least three important decisions remain. First, there is the issue of curve smoothing. Applying a smoothing procedure can help to reveal a curve's shape when partially obscured by sampling error, yet it also raises the risk of flattening genuine peaks. A deliberate choice of whether or not to smooth should be based on a consideration of this tradeoff, perhaps informed by consulting results from comparison data. The MAXEIG program always plots the raw data points, along with either a LOWESS (use `Gr.Smooth = 1`) or running medians (use `Gr.Smooth = 2`) smoothed curve if requested (for no smoothing, use `Gr.Smooth = 0`; this is the default). Second, you need to decide whether to apply the same y scale to all curves (use `Gr.Common.Y = T`) or allow scales to vary (use `Gr.Common.Y = F`, which is the default). It has been suggested that a common scale should be used to prevent apparent peaks to emerge in a subset of curves through an exaggeration of sampling error. However, the variables themselves may have been assessed using different units of measurement (e.g., a scale ranging from 0-20 vs. *T* score units) and may be of differential validity. Thus, there may be little reason to expect covariances or eigenvalues to reach comparable heights across curves. One useful approach may be to standardize all indicators prior to analysis (use `St.Ind = T`) and then apply a common y scale; this at least equates the variances of the indicators, though not their validities. Third, you must choose whether to interpret the full panel of curves or an averaged curve. It has not been determined which approach maximizes the accuracy of structural inferences. The program will provide the full panel of curves even if you also opt for the averaged curve.

In addition to these decisions, there are a couple of points that you should know regarding calculation methods. The General Covariance Mixture Theorem is used to estimate the proportion of taxon members within each subsample, and the overall taxon base rate, for each curve. (Note that this is done regardless of whether covariances or eigenvalues are used as the measure of association.) One can choose how to assign cases to the taxon and complement for the estimation of additional latent parameters. These class assignments can be saved for subsequent analysis (see below), and they are also used in the further estimation of latent class *Ms* and *SDs*, as well as indicator validity. The use of Bayes' Theorem to assign cases to latent classes requires the estimation of valid and false positive rates achieved by the hitmax cut on each indicator, which in turn can only be estimated when individual variables serve as the input indicator (as opposed to composite input indicators). If this condition is not met, the program assigns cases to latent classes using the grand base rate estimate. That is, based on the total score on all variables submitted for analysis, the highest-scoring cases are assigned to the taxon, and the lowest-scoring cases are assigned to the complement. The proportion of cases assigned to the taxon equals the estimated taxon base rate. The base-rate classification is the default method; setting `Classify = 2` requests the use of Bayes' Theorem, though as noted above the program will automatically revert to base-rate classification if composite input indicators are used. In any event, the classification method is reported in the output.

Finally, a brief note is in order on saving the class assignments or Bayesian probabilities of taxon membership calculated by the program. If you wish to save either of these values for subsequent analysis, you must not only assign `Save.Class` a value of 1 or 2, respectively, but also assign the program call to a variable that will store the classification values (e.g., `Classes <- MAXEIG(TestCat[,1:4], Save.Class = 1)`). Classification codes are returned as a vector of 1's (cases assigned to the complement class) and 2's (cases assigned to the taxon class) in the order in which cases appeared in the original data file, which should facilitate merging the classification results with the original data for subsequent analyses (e.g., the command line `TestCat <- cbind(TestCat, Classes)` would add the saved classification codes from the MAXEIG analysis as a final column in `TestCat`. Likewise, Bayesian probabilities are returned as a vector of values in the order in which cases appeared in the original data file.

The following sample commands will conduct MAXEIG analyses (1) of `TestCat`, using all program defaults except that comparison data are not used; (2) of `TestCat`, this time using the comparison data; and (3) of `SkewedDim`, using 10 internal replications (to reduce the obfuscating influence of assigning equal-scoring cases to different windows) and comparison data:

```
MAXEIG(TestCat[,1:4], Comp.Data = F)
MAXEIG(TestCat[,1:4])
MAXEIG(SkewedDim[,1:4], Replications = 10)
```

8.6 Output:

As described above, the program provides a vector containing class assignments if so requested. The program plots a full panel of curves and, if requested, a panel of curves averaged for each input indicator and/or a single averaged curve. Each set of graphs appears in its own labeled window, provided that output is not restricted to the 2-panel plot including results for comparison data. Each curve is a plot of either covariances or eigenvalues along the y axis by input subsamples along the x axis, with axes and values fully labeled according to the methods of allocating variables to the input/output roles and of dividing cases into subsamples. If comparison data are analyzed, a 2-panel graph sheet presents the results for the research data (dark line with points) superimposed

above the results for the categorical comparison data, and then the results for the dimensional comparison data; results for comparison data sets are summarized by plotting the middle 50% of data points as a gray band and light lines that show the minimum and maximum values.

The text summary contains a wealth of information. First, a detailed overview of relevant data parameters and program specifications is provided. This includes the method by which variables were assigned to their input/output roles, the number and type of subsamples that were formed along the input, the number of internal replications, whether smoothed values were used for graphing and estimation, how latent parameters were estimated, how cases were assigned to groups for the estimation of latent parameters, and so forth.

This procedural summary is followed by several statistical summaries. First is the estimated hitmax location and base rate estimate derived from each curve (listed in the order in which the curves are plotted) and a summary (M , SD) of estimates across curves. In cases in which each indicator served as input for more than one curve, base rate estimates are then collapsed for each indicator variable and followed by a summary (M , SD) of estimates across indicators. If curves were averaged, the base rate estimate(s) for the averaged curve(s) are provided (if the inchworm consistency test was used there will be an averaged curve and base rate estimate for each number of windows). This is followed by all of the output from the `Indicator.Dist` procedure (described above). If comparison data are generated and analyzed, a summary of the accuracy with which correlations were reproduced for each data set is provided along with the M s and SD s of the base rate estimates for each sample of comparison data and a curve fit index (CCFI; see Chapter 4).

Finally, note that all output is labeled as either MAXCOV or MAXEIG depending on whether covariances or eigenvalues were calculated. Sample output for the `TestCat` and `SkewedDim` analyses (commands above) appear below (both are MAXEIG analyses).

```
> MAXEIG(TestCat[,1:4])
```

SUMMARY OF MAXEIG ANALYTIC SPECIFICATIONS

```
Sample size: 1000
Number of indicator variables: 4
Replications: 1
Subsamples: 25 windows with 0.9 overlap
n per window at 25 windows: 294
Indicators: Each variable serves once as input, with all other variables as outputs
Total number of curves: 4
Y values smoothed for graphing and estimation: No
Base rate estimation: Adapted general covariance mixture theorem
Classification of cases: Cut total score at estimated base rate
Size of finite populations of comparison data: 1e+05
Number of samples of comparison data drawn from each population: 100
```

SUMMARY OF MAXEIG PARAMETER ESTIMATES

Estimated hitmax values and taxon base rates for each curve:

	Hitmax	P
Curve 1	0.701	0.194
Curve 2	0.789	0.217
Curve 3	0.712	0.217
Curve 4	0.759	0.232

Summary of base rate estimates across curves:

```
M = 0.215
SD = 0.016
```

Estimated VP, FP values at each indicator's hitmax cut:

	VP	FP
Indicator 1	0.5065941	0.07651047
Indicator 2	0.3531375	0.05545790
Indicator 3	0.4627956	0.07610102
Indicator 4	0.4645831	0.06817386

[continued on the next page]

Base rate estimate for averaged curve = 0.214

Indicator distributions in the full sample (N = 1000):

	M	SD	Skew	Kurtosis
Ind 1	0	1	0.298	-0.053
Ind 2	0	1	0.407	0.037
Ind 3	0	1	0.293	-0.266
Ind 4	0	1	0.307	-0.208
M	0	1	0.326	-0.122
SD	0	0	0.054	0.139

Indicator distributions in the taxon (n = 215):

	M	SD	Skew	Kurtosis
Ind 1	1.244	0.742	-0.170	0.520
Ind 2	1.266	0.734	0.122	-0.072
Ind 3	1.225	0.680	0.031	-0.460
Ind 4	1.210	0.723	-0.012	0.490
M	1.236	0.720	-0.007	0.120
SD	0.024	0.028	0.122	0.473

Indicator distributions in the complement (n = 785):

	M	SD	Skew	Kurtosis
Ind 1	-0.341	0.764	-0.124	-0.039
Ind 2	-0.347	0.753	0.048	0.272
Ind 3	-0.335	0.790	0.139	0.145
Ind 4	-0.331	0.787	0.082	-0.158
M	-0.339	0.774	0.036	0.055
SD	0.007	0.018	0.113	0.191

Between-group validity:

	Raw Units	Cohen's d
Ind 1	1.585	2.088
Ind 2	1.613	2.152
Ind 3	1.560	2.032
Ind 4	1.542	1.992
M	1.575	2.066
SD	0.031	0.070

Indicator correlations:

Full Sample (N = 1000):

	Ind 1	Ind 2	Ind 3	Ind 4
Ind 1	1.000	0.433	0.440	0.452
Ind 2	0.433	1.000	0.446	0.415
Ind 3	0.440	0.446	1.000	0.433
Ind 4	0.452	0.415	0.433	1.000

Taxon (n = 215):

	Ind 1	Ind 2	Ind 3	Ind 4
Ind 1	1.000	-0.071	-0.130	-0.113
Ind 2	-0.071	1.000	-0.045	-0.174
Ind 3	-0.130	-0.045	1.000	-0.174
Ind 4	-0.113	-0.174	-0.174	1.000

Complement (n = 785):

	Ind 1	Ind 2	Ind 3	Ind 4
Ind 1	1.000	0.021	0.076	0.111
Ind 2	0.021	1.000	0.054	0.033
Ind 3	0.076	0.054	1.000	0.093
Ind 4	0.111	0.033	0.093	1.000

[continued on the next page]

Summary of indicator correlations:

	M	SD
Full Sample	0.437	0.013
Taxon	-0.118	0.053
Complement	0.065	0.035

Generating and analyzing comparison data

Categorical population generated, RMSR $r = 0.011$

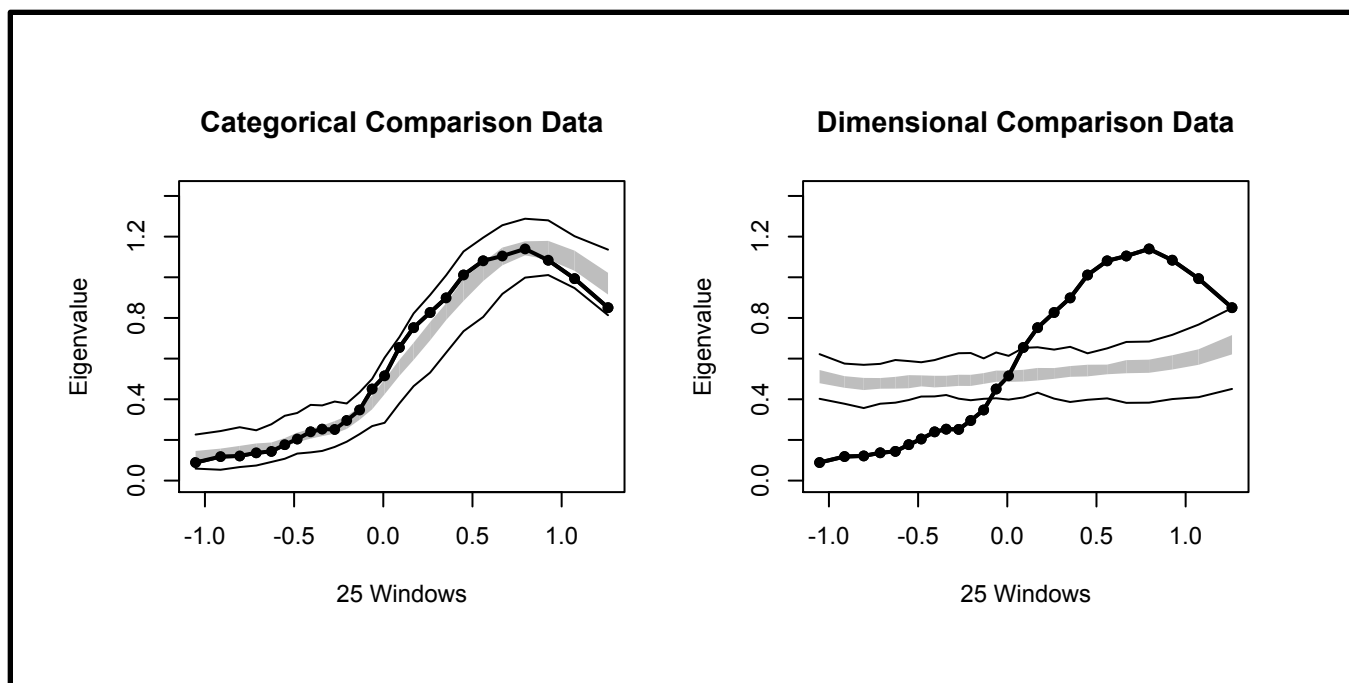
Analysis of 100 samples of categorical comparison data completed

Dimensional population generated, RMSR $r = 0.01$

Analysis of 100 samples of dimensional comparison data completed

Comparison curve fit index (CCFI) = $0.3456 / (0.3456 + 0.0824) = 0.807$

Note: CCFI values can range from 0 (dimensional) to 1 (categorical). The more a CCFI value deviates from .50, the stronger the result. When $.40 < \text{CCFI} < .60$, this should be interpreted with caution.



> MAXEIG(SkewedDim[,1:4])

SUMMARY OF MAXEIG ANALYTIC SPECIFICATIONS

Sample size: 600
Number of indicator variables: 4
Replications: 10
Subsamples: 25 windows with 0.9 overlap
n per window at 25 windows: 176
Indicators: Each variable serves once as input, with all other variables as outputs
Total number of curves: 4
Y values smoothed for graphing and estimation: No
Base rate estimation: Adapted general covariance mixture theorem
Classification of cases: Cut total score at estimated base rate
Size of finite populations of comparison data: 1e+05
Number of samples of comparison data drawn from each population: 100

SUMMARY OF MAXEIG PARAMETER ESTIMATES

Estimated hitmax values and taxon base rates for each curve:

	Hitmax	P
Curve 1	1.351	0.237
Curve 2	1.396	0.240
Curve 3	0.775	0.235
Curve 4	0.593	0.287

Summary of base rate estimates across curves:

M = 0.25
SD = 0.025

Estimated VP, FP values at each indicator's hitmax cut:

	VP	FP
Indicator 1	0.08405114	0.02616697
Indicator 2	0.08318455	0.02625211
Indicator 3	0.08475397	0.02609959
Indicator 4	0.06947844	0.02799499

Base rate estimate for averaged curve = 0.249

Indicator distributions in the full sample (N = 600):

	M	SD	Skew	Kurtosis
Ind 1	0	1	0.895	0.431
Ind 2	0	1	0.858	0.323
Ind 3	0	1	0.943	0.781
Ind 4	0	1	0.948	0.715
M	0	1	0.911	0.562
SD	0	0	0.043	0.221

Indicator distributions in the taxon (n = 150):

	M	SD	Skew	Kurtosis
Ind 1	0.938	1.039	0.198	-0.423
Ind 2	1.048	1.006	0.128	-0.514
Ind 3	1.049	1.024	0.385	-0.223
Ind 4	0.963	1.041	0.494	-0.543
M	0.999	1.028	0.301	-0.426
SD	0.058	0.016	0.168	0.145

[continued on the next page]

Indicator distributions in the complement (n = 450):

	M	SD	Skew	Kurtosis
Ind 1	-0.313	0.764	0.927	0.875
Ind 2	-0.349	0.713	0.645	-0.207
Ind 3	-0.350	0.704	0.576	-0.341
Ind 4	-0.321	0.749	0.749	0.311
M	-0.333	0.733	0.724	0.160
SD	0.019	0.029	0.153	0.554

Between-group validity:

	Raw Units	Cohen's d
Ind 1	1.250	1.487
Ind 2	1.397	1.754
Ind 3	1.399	1.758
Ind 4	1.284	1.544
M	1.333	1.636
SD	0.077	0.141

Indicator correlations:

Full Sample (N = 600):

	Ind 1	Ind 2	Ind 3	Ind 4
Ind 1	1.000	0.396	0.395	0.283
Ind 2	0.396	1.000	0.393	0.388
Ind 3	0.395	0.393	1.000	0.339
Ind 4	0.283	0.388	0.339	1.000

Taxon (n = 150):

	Ind 1	Ind 2	Ind 3	Ind 4
Ind 1	1.000	-0.012	0.053	-0.191
Ind 2	-0.012	1.000	-0.086	-0.068
Ind 3	0.053	-0.086	1.000	-0.152
Ind 4	-0.191	-0.068	-0.152	1.000

Complement (n = 450):

	Ind 1	Ind 2	Ind 3	Ind 4
Ind 1	1.000	0.174	0.130	0.077
Ind 2	0.174	1.000	0.127	0.172
Ind 3	0.130	0.127	1.000	0.106
Ind 4	0.077	0.172	0.106	1.000

Summary of indicator correlations:

	M	SD
Full Sample	0.366	0.046
Taxon	-0.076	0.089
Complement	0.131	0.037

Generating and analyzing comparison data

Categorical population generated, RMSR r = 0.017

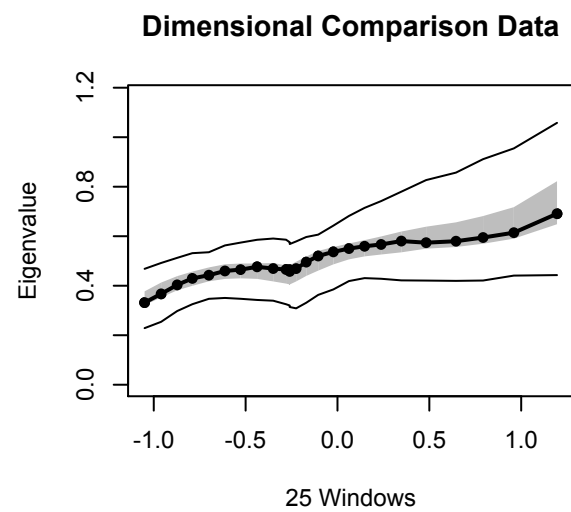
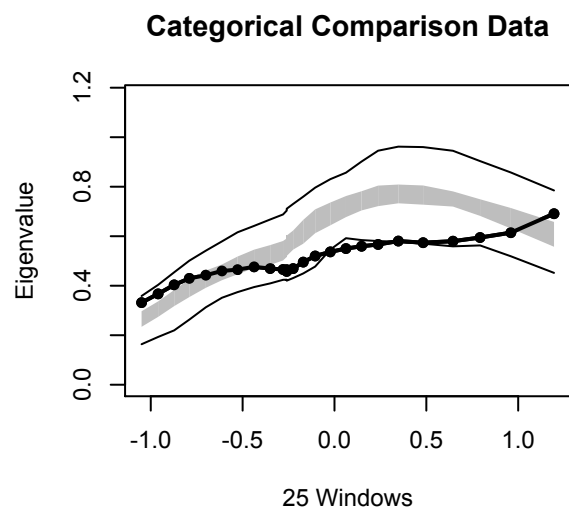
Analysis of 100 samples of categorical comparison data completed

Dimensional population generated, RMSR r = 0.022

Analysis of 100 samples of dimensional comparison data completed

Comparison curve fit index (CCFI) = $0.0106 / (0.0106 + 0.1134) = 0.086$

Note: CCFI values can range from 0 (dimensional) to 1 (categorical). The more a CCFI value deviates from .50, the stronger the result. When $.40 < \text{CCFI} < .60$, this should be interpreted with caution.



Chapter 9: L-Mode

Performing Latent Mode (L-Mode) Taxometric Analyses

9.1 Overview:

This program runs the latent mode (L-Mode) taxometric procedure introduced by Waller and Meehl (1998).

9.2 Command:

```
LMode <- function(Data.Set, Comp.Data = T, N.Pop = 100000, N.Samples = 100, Supplied.Class = F, Supplied.P = 0,
  Mode.L = -.001, Mode.R = .001, St.Ind = T, Gr.Comp.Only = T, Classify = 1, Save.Class = F,
  File.Output = F, File.Name = "Output.txt", Seed = 1)
```

9.3 Required Arguments:

Data.Set – The supplied data file, which must contain one case per row, one indicator per column. If a variable is included to signify group membership for each case (1 = complement, 2 = taxon), this must be in the final column of the data provided to the program (see **Supplied.Class**, below). Cases missing any data will be removed prior to analysis.

9.4 Optional Arguments:

Comp.Data – Whether to generate and analyze categorical and dimensional comparison data (default = T). When this is set to T, comparison data are used (see Chapter 4 for details) and, because this involves the averaging of curves, indicators are standardized.

N.Pop – The size of the finite populations of categorical and dimensional comparison data (default = 100,000). Unless the number of indicators is unusually large, this should run reasonably quickly.

N.Samples – Number of comparison data sets of each structure to generate and analyze (default = 100). Generating multiple sets of comparison data is strongly encouraged, as it allows one to examine a sampling distribution of results for each structure.

Supplied.Class – Whether the final column of the supplied data set contains group membership (coded as 1 = complement, 2 = taxon) to use for the estimation of data parameters and the generation of categorical comparison data (default = F).

Supplied.P – As an alternative to providing group membership, the program will accept a user-specified taxon base rate and assign cases to groups using the base-rate classification method described in Chapter 4 (default = 0). If this value is left at 0 *and* group membership is not provided, the program checks the value of **Classify** to determine how to assign cases to groups (see below).

Mode.L – Position beyond which to search for the left mode (default = -.001). Note that, for a particularly high base rate taxon, you may need to specify a value that corresponds to a visible trough in the L-Mode curve (obtained by first running the program with the default value of **Mode.L**) for the program to properly locate a clearly visible mode to the left of zero.

Mode.R – Position beyond which to search for the right mode (default = .001). Note that, for a particularly low base rate taxon, you may need to specify a value that corresponds to a visible trough in the L-Mode curve (obtained by first running the program with the default value of **Mode.L**) for the program to properly locate a clearly visible mode to the right of zero.

St.Ind – Whether to standardize each indicator prior to analysis (default = T).

Gr.Comp.Only – Whether to restrict graphical output to the 2-panel plot that includes results for comparison data (default = T).

Classify – Whenever the user does not supply a taxon base rate or a classification of cases, this parameter determines whether to classify cases using the base-rate method (1, the default value) or the nearest mode (2).

Save.Class – Whether to save class assignments (1 = complement, 2 = taxon) calculated by program (default = F).

File.Output – Whether to send text output to a file rather than displaying it on the screen (default = F).

File.Name – The filename to use when text output is sent to a file (default = "Output.txt"). Note that (1) this has no effect if **File.Output** = F and (2) this file appears in your default R directory unless you specify a full path along with the filename.

Seed – Random number seed provided prior to analysis of empirical data as well as prior to generating each population of comparison data (if comparison data are used); the default value is 1. In addition to affording exact replications of analyses, this enables the user to generate and analyze identical populations and samples of comparison data across analyses using different taxometric procedures.

To ensure that identical populations and samples are used, make sure that `N.Pop` and `N.Samples` are held constant across analyses and that the same classification of cases is used to generate categorical comparison data. This latter requirement can be achieved by providing classification codes (and setting `Supplied.Class = T`) or by providing the same taxon base rate estimate for base-rate classification (setting `Supplied.P` to the same value for each analysis).

9.5 Running the Program:

This is perhaps the simplest of the taxometric procedures to conduct, as nearly all aspects are automated and the analysis only involves one graph. As with all of the programs, care must be taken in submitting the data for analysis.

One unique aspect of the L-Mode program that requires careful attention is the specification of regions to search for the left and right modes. For example, a low base rate taxon may yield a small peak far to the right of zero, in which case the correspondingly large complement class may yield a taller peak just to the left of zero. Thus, the taxon mode may be missed if the search includes values slightly to the right of zero, as the complement distribution can run past zero at a greater height. (The reverse is true for high base rate taxa.) Thus, after first running L-Mode using the default values of `Mode.L` and `Mode.R`, it would be wise to visually check whether the program has correctly located the left and right modes. There is no foolproof way to automate this process, so be sure to specify a value that approximate a visible trough in the distribution so that the program's search can proceed from there. For example, with a small base rate taxon, it may be useful to set `Mode.R = 1` (or possibly even higher).

As in MAXEIG, you must assign the program call to a variable if you want to save class assignments for subsequent analysis (see section 9.5). Finally, note that the R language does not allow the usual hyphen in "L-Mode." The following commands would conduct L-Mode analyses of `TestCat` and `SkewedDim`, respectively (note that first command was run initially without specifying `Mode.R`, but the re-run with `Mode.R` set to 1 to correctly locate the right-hand mode that is clearly visible in the graph; this is an example of how a large left-hand mode can cause the height of the curve at $x = 0$ to exceed that of the height of the right-hand mode):

```
Lmode(TestCat[,1:4], Mode.R = 1)
Lmode(SkewedDim)
```

9.6 Output:

As noted above, the program returns a vector of class assignments if so requested (see section 9.5 for how to save and use these). The program plots an L-Mode graph consisting of the frequency distribution of estimated scores on the first factor of a factor analysis, with vertical lines representing the two estimated latent modes (unless output is restricted to the 2-panel plot described next). If comparison data are analyzed, a 2-panel graph sheet presents the results for the research data (dark line) superimposed above the results for the categorical comparison data, and then the results for the dimensional comparison data; results for comparison data sets are summarized by plotting the middle 50% of data points as a gray band and light lines that show the minimum and maximum values.

In addition, a text summary includes an analytic overview, base rate estimates derived from the location of each mode (plus their average), taxon and complement means on each indicator estimated via factor loadings, a base rate estimate derived from the classification of cases, all of the output from the `Indicator.Dist` procedure (see Chapter 5), and estimated validity for the factor scores (in "raw" and standardized—using Cohen's d —units). If comparison data are generated and analyzed, a summary of the accuracy with which correlations were reproduced for each data set is provided (see Chapter 4). Sample output for the `TestCat` and `SkewedDim` analyses (commands above) appear below.

```
> LMode(TestCat[,1:4], Mode.R = 1)
```

SUMMARY OF L-MODE ANALYTIC SPECIFICATIONS

Sample size: 1000
Number of indicator variables: 4
Classification of cases: Cut total score at estimated base rate
Size of finite populations of comparison data: 1e+05
Number of samples of comparison data drawn from each population: 100

SUMMARY OF L-MODE PARAMETER ESTIMATES

Summary of taxon base rate estimates:
Based on location of left mode: 0.271
Based on location of right mode: 0.303
M = 0.287

Estimated latent group M on each indicator (via factor loadings):

		Taxon	Complement
Indicator 1	1.060	-0.426	
Indicator 2	1.024	-0.411	
Indicator 3	1.053	-0.423	
Indicator 4	1.032	-0.415	

Indicator distributions in the full sample (N = 1000):

	M	SD	Skew	Kurtosis
Ind 1	0	1	0.298	-0.053
Ind 2	0	1	0.407	0.037
Ind 3	0	1	0.293	-0.266
Ind 4	0	1	0.307	-0.208
M	0	1	0.326	-0.122
SD	0	0	0.054	0.139

Indicator distributions in the taxon (n = 287):

	M	SD	Skew	Kurtosis
Ind 1	1.040	0.784	0.027	-0.036
Ind 2	1.039	0.821	0.046	-0.208
Ind 3	1.053	0.743	-0.020	-0.341
Ind 4	1.061	0.756	-0.110	0.587
M	1.048	0.776	-0.014	0.001
SD	0.011	0.035	0.070	0.410

Indicator distributions in the complement (n = 713):

	M	SD	Skew	Kurtosis
Ind 1	-0.418	0.739	-0.130	0.020
Ind 2	-0.418	0.722	-0.025	0.253
Ind 3	-0.424	0.744	0.053	0.111
Ind 4	-0.427	0.733	0.010	-0.044
M	-0.422	0.735	-0.023	0.085
SD	0.004	0.009	0.078	0.129

Between-group validity:

	Raw Units	Cohen's d
Ind 1	1.458	1.939
Ind 2	1.457	1.937
Ind 3	1.478	1.986
Ind 4	1.488	2.012
M	1.470	1.969
SD	0.015	0.036

Between-group validity on factor scores:

Raw Units	Cohen's d
1.934	4.018

[continued on the next page]

Indicator correlations:

Full Sample (N = 1000):

	Ind 1	Ind 2	Ind 3	Ind 4
Ind 1	1.000	0.433	0.440	0.452
Ind 2	0.433	1.000	0.446	0.415
Ind 3	0.440	0.446	1.000	0.433
Ind 4	0.452	0.415	0.433	1.000

Taxon (n = 287):

	Ind 1	Ind 2	Ind 3	Ind 4
Ind 1	1.000	0.127	0.060	0.062
Ind 2	0.127	1.000	0.141	-0.010
Ind 3	0.060	0.141	1.000	-0.079
Ind 4	0.062	-0.010	-0.079	1.000

Complement (n = 713):

	Ind 1	Ind 2	Ind 3	Ind 4
Ind 1	1.000	-0.068	-0.029	-0.007
Ind 2	-0.068	1.000	-0.052	-0.071
Ind 3	-0.029	-0.052	1.000	-0.011
Ind 4	-0.007	-0.071	-0.011	1.000

Summary of indicator correlations:

	M	SD
Full Sample	0.437	0.013
Taxon	0.050	0.083
Complement	-0.040	0.028

Generating and analyzing comparison data

Categorical population generated, RMSR r = 0.01

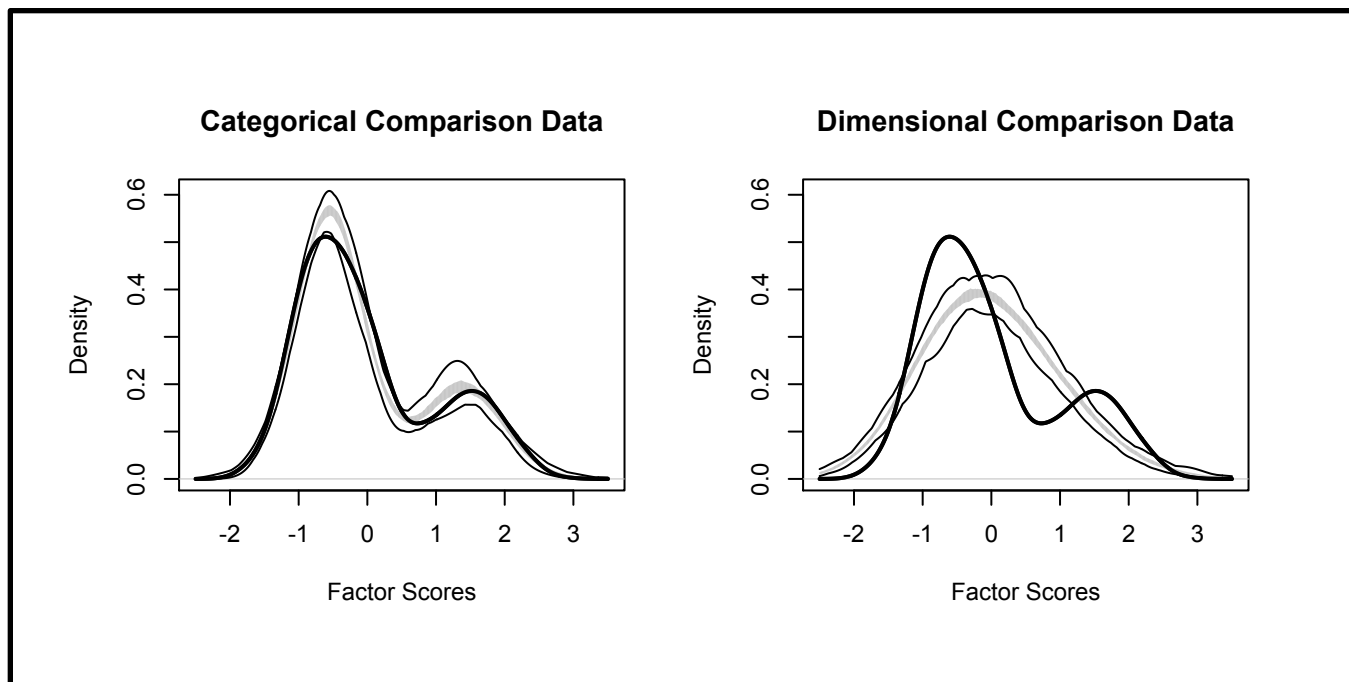
Analysis of 100 samples of categorical comparison data completed

Dimensional population generated, RMSR r = 0.01

Analysis of 100 samples of dimensional comparison data completed

Comparison curve fit index (CCFI) = $0.0464 / (0.0464 + 0.0064) = 0.878$

Note: CCFI values can range from 0 (dimensional) to 1 (categorical). The more a CCFI value deviates from .50, the stronger the result. When $.40 < \text{CCFI} < .60$, this should be interpreted with caution.



> LMode(SkewedDim)

SUMMARY OF L-MODE ANALYTIC SPECIFICATIONS

Sample size: 600
Number of indicator variables: 4
Classification of cases: Cut total score at estimated base rate
Size of finite populations of comparison data: 1e+05
Number of samples of comparison data drawn from each population: 100

SUMMARY OF L-MODE PARAMETER ESTIMATES

Summary of taxon base rate estimates:
Based on location of left mode: 0.258
Based on location of right mode: 1
M = 0.629

Estimated latent group M on each indicator (via factor loadings):

		Taxon	Complement
Indicator 1	-0.452	0.766	
Indicator 2	-0.514	0.871	
Indicator 3	-0.480	0.814	
Indicator 4	-0.414	0.701	

Indicator distributions in the full sample (N = 600):

	M	SD	Skew	Kurtosis
Ind 1	0	1	0.895	0.431
Ind 2	0	1	0.858	0.323
Ind 3	0	1	0.943	0.781
Ind 4	0	1	0.948	0.715
M	0	1	0.911	0.562
SD	0	0	0.043	0.221

Indicator distributions in the taxon (n = 377):

	M	SD	Skew	Kurtosis
Ind 1	0.425	0.969	0.715	0.068
Ind 2	0.408	0.985	0.590	-0.013
Ind 3	0.414	0.972	0.785	0.653
Ind 4	0.416	0.982	0.741	0.393
M	0.416	0.977	0.708	0.275
SD	0.007	0.008	0.083	0.307

Indicator distributions in the complement (n = 223):

	M	SD	Skew	Kurtosis
Ind 1	-0.718	0.532	0.901	0.273
Ind 2	-0.690	0.541	1.039	0.916
Ind 3	-0.699	0.562	1.245	1.149
Ind 4	-0.703	0.526	1.012	1.246
M	-0.702	0.540	1.049	0.896
SD	0.012	0.016	0.143	0.438

Between-group validity:

	Raw Units	Cohen's d
Ind 1	1.143	1.370
Ind 2	1.098	1.295
Ind 3	1.113	1.319
Ind 4	1.118	1.328
M	1.118	1.328
SD	0.019	0.031

Between-group validity on factor scores:

Raw Units	Cohen's d
1.538	2.458

[continued on the next page]

Indicator correlations:

Full Sample (N = 600):

	Ind 1	Ind 2	Ind 3	Ind 4
Ind 1	1.000	0.396	0.395	0.283
Ind 2	0.396	1.000	0.393	0.388
Ind 3	0.395	0.393	1.000	0.339
Ind 4	0.283	0.388	0.339	1.000

Taxon (n = 377):

	Ind 1	Ind 2	Ind 3	Ind 4
Ind 1	1.000	0.157	0.193	0.010
Ind 2	0.157	1.000	0.209	0.188
Ind 3	0.193	0.209	1.000	0.079
Ind 4	0.010	0.188	0.079	1.000

Complement (n = 223):

	Ind 1	Ind 2	Ind 3	Ind 4
Ind 1	1.000	0.078	-0.150	-0.208
Ind 2	0.078	1.000	-0.167	-0.130
Ind 3	-0.150	-0.167	1.000	0.005
Ind 4	-0.208	-0.130	0.005	1.000

Summary of indicator correlations:

	M	SD
Full Sample	0.366	0.046
Taxon	0.140	0.079
Complement	-0.095	0.111

Generating and analyzing comparison data

Categorical population generated, RMSR r = 0.029

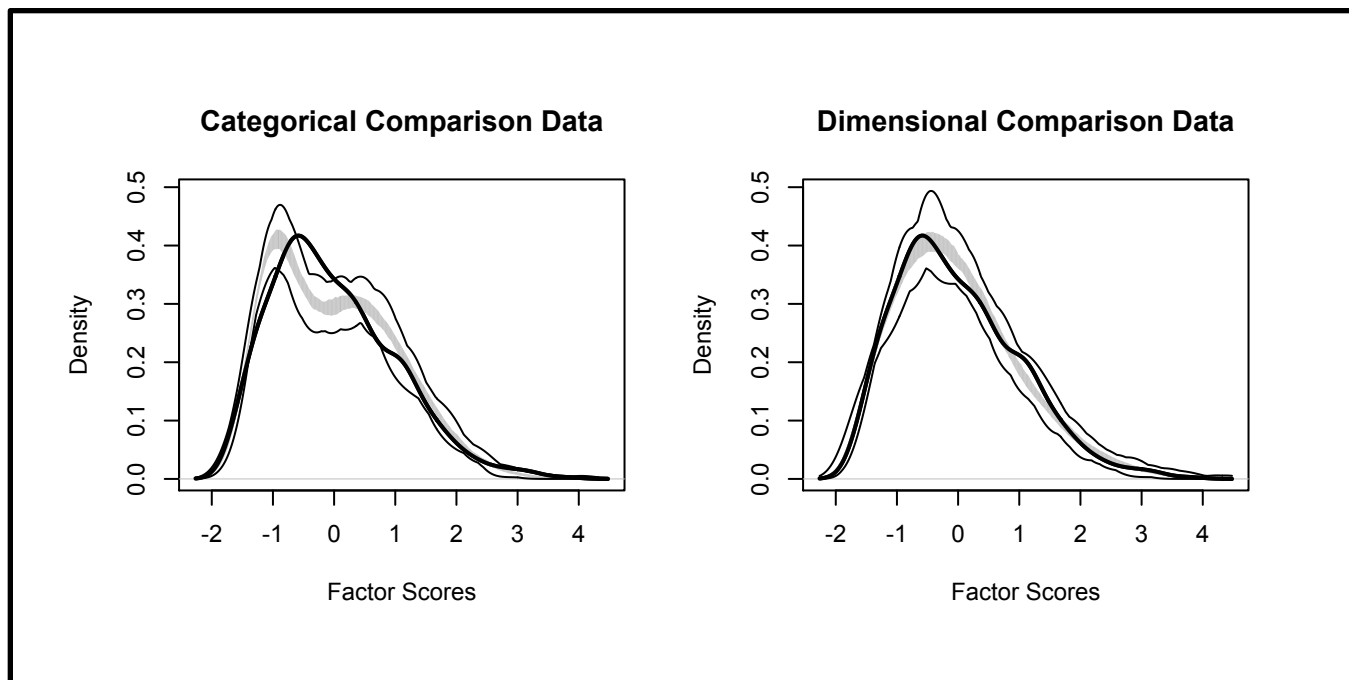
Analysis of 100 samples of categorical comparison data completed

Dimensional population generated, RMSR r = 0.022

Analysis of 100 samples of dimensional comparison data completed

Comparison curve fit index (CCFI) = $0.005 / (0.005 + 0.0119) = 0.295$

Note: CCFI values can range from 0 (dimensional) to 1 (categorical). The more a CCFI value deviates from .50, the stronger the result. When $.40 < \text{CCFI} < .60$, this should be interpreted with caution.



Chapter 10: MAXSLOPE

Performing Maximum Slope (MAXSLOPE) Taxometric Analyses

10.1 Overview:

This program runs the maximum slope (MAXSLOPE) taxometric procedure introduced by Grove and Meehl (1993), discussed in greater detail by Grove (2004), and studied by Ruscio and Walters (2011); the latter paper suggests that MAXSLOPE might be a useful adjunct to MAMBAC when only two indicators are available for analysis. When more than two indicators are available, MAXCOV or MAXEIG can be performed instead of MAXSLOPE.

10.2 Command:

```
MAXSLOPE <- function(Data.Set, Comp.Data = T, N.Pop = 100000, N.Samples = 100, Supplied.Class = F,  
                      Supplied.P = 0, File.Output = F, File.Name = "Output.txt", Seed = 1)
```

10.3 Required Arguments:

Data.Set – The supplied data file, which must contain one case per row, one indicator per column. If a variable is included to signify group membership for each case (1 = complement, 2 = taxon), this must be in the final column of the data provided to the program (see **Supplied.Class**, below). Cases missing any data will be removed prior to analysis. If more than two indicators are provided, only the first two indicators will be analyzed and a message to this effect will appear in the text output.

10.4 Optional Arguments:

Comp.Data – Whether to generate and analyze categorical and dimensional comparison data (default = T). When this is set to T, comparison data are used (see Chapter 4 for details).

N.Pop – The size of the finite populations of categorical and dimensional comparison data (default = 100,000). Unless the number of indicators is unusually large, this should run reasonably quickly.

N.Samples – Number of comparison data sets of each structure to generate and analyze (default = 100). Generating multiple sets of comparison data is strongly encouraged, as it allows one to examine a sampling distribution of results for each structure.

Supplied.Class – Whether the final column of the supplied data set contains group membership (coded as 1 = complement, 2 = taxon) to use for the estimation of data parameters and the generation of categorical comparison data (default = F).

Supplied.P – As an alternative to providing group membership, the program will accept a user-specified taxon base rate and assign cases to groups using the base-rate classification method described in Chapter 4 (default = 0). If this value is left at 0 *and* group membership is not provided, the program checks the value of **Classify** to determine how to assign cases to groups (see below).

File.Output – Whether to send text output to a file rather than displaying it on the screen (default = F).

File.Name – The filename to use when text output is sent to a file (default = “Output.txt”). Note that (1) this has no effect if **File.Output** = F and (2) this file appears in your default R directory unless you specify a full path along with the filename.

Seed – Random number seed provided prior to analysis of empirical data as well as prior to generating each population of comparison data (if comparison data are used); the default value is 1. In addition to affording exact replications of analyses, this enables the user to generate and analyze identical populations and samples of comparison data across analyses using different taxometric procedures. To ensure that identical populations and samples are used, make sure that **N.Pop** and **N.Samples** are held constant across analyses and that the same classification of cases is used to generate categorical comparison data. This latter requirement can be achieved by providing classification codes (and setting **Supplied.Class** = T) or by providing the same taxon base rate estimate for base-rate classification (setting **Supplied.P** to the same value for each analysis).

10.5 Running the Program:

Like L-Mode, this is a very simple taxometric procedure to conduct because nearly all aspects are automated and the analysis only involves one graph. As with all of the programs, care must be taken in submitting the data for analysis.

A few notes on the present implementation of MAXSLOPE are in order. This program does not display indicator scatterplots. Instead, the graphs are of the slopes generated by a locally weighted scatterplot smoother (LOWESS; Cleveland, 1979) applied to the scatterplots. Likewise, the CCFI is calculated using these slopes, not data points in the original scatterplots; for comparison data, the median (rather than mean) slope is used at each x value because slopes occasionally equal positive or negative infinity, hence a mean cannot be calculated. The program only performs MAXSLOPE using two indicators. If more than two indicators are available, one could perform either MAXCOV/MAXEIG or perform MAXSLOPE using all possible indicator pairs; the latter choice would require running this program multiple times. Finally, the program averages results across the two MAXSLOPE curves that are generated by swapping the two indicators in the roles as x and y variables in a scatterplot to generate slopes.

The following commands would conduct MAXSLOPE analyses of `TestCat` and `SkewedDim`, respectively, using only the first two indicators from each data set:

```
MAXSLOPE (TestCat[,1:2])
MAXSLOPE (SkewedDim[,1:2])
```

10.6 Output:

The program plots a MAXSLOPE graph consisting of the slopes from a locally weighted scatterplot smoother. Two curves are generated by using the indicators in both x - y configurations, and the slopes are averaged. (Because slopes are averaged, each indicator is standardized prior to analysis.) If comparison data are analyzed, a 2-panel graph sheet presents the results for the research data (dark line) superimposed above the results for the categorical comparison data, and then the results for the dimensional comparison data; results for comparison data sets are summarized by plotting the middle 50% of data points as a gray band and light lines that show the minimum and maximum values.

In addition, a text summary includes an analytic overview, a base rate estimate calculated from the averaged MAXSLOPE curve (specifically, the proportion of cases above the point of maximum slope), and all of the output from the `Indicator.Dist` procedure (see Chapter 5). If comparison data are generated and analyzed, a summary of the accuracy with which correlations were reproduced for each data set is provided (see Chapter 4). Sample output for the `TestCat` and `SkewedDim` analyses (commands above) appear below. Note that the graphs for the `SkewedDim` analysis take on relatively few unique x values because the data vary along ordered categorical scales. Nonetheless, the results for these data were clearly more similar to those for the dimensional than the categorical comparison data, and both a visual inspection and the CCFI correctly identified dimensional structure.

```
> MAXSLOPE (TestCat[,1:2])
```

SUMMARY OF MAXSLOPE ANALYTIC SPECIFICATIONS

```
Sample size: 1000
Classification of cases: Cut total score at estimated base rate
Size of finite populations of comparison data: 1e+05
Number of samples of comparison data drawn from each population: 100
```

SUMMARY OF MAXSLOPE PARAMETER ESTIMATES

```
Estimated taxon base rate = 0.242
```

```
Indicator distributions in the full sample ( N = 1000 ):
```

	M	SD	Skew	Kurtosis
Ind 1	0	1	0.298	-0.053
Ind 2	0	1	0.407	0.037
M	0	1	0.352	-0.008
SD	0	0	0.077	0.064

```
Indicator distributions in the taxon ( n = 242 ):
```

	M	SD	Skew	Kurtosis
Ind 1	1.212	0.700	0.130	0.209
Ind 2	1.238	0.718	0.201	-0.041
M	1.225	0.709	0.166	0.084
SD	0.018	0.012	0.050	0.177

```
[continued on the next page]
```

Indicator distributions in the complement (n = 758):

	M	SD	Skew	Kurtosis
Ind 1	-0.387	0.738	-0.174	-0.012
Ind 2	-0.395	0.714	-0.104	0.146
M	-0.391	0.726	-0.139	0.067
SD	0.006	0.017	0.050	0.112

Between-group validity:

	Raw Units	Cohen's d
Ind 1	1.599	2.194
Ind 2	1.633	2.285
M	1.616	2.240
SD	0.024	0.065

Indicator correlations:

Full Sample $r = 0.433$
Taxon $r = -0.124$
Complement $r = -0.079$

Generating and analyzing comparison data

Categorical population generated, RMSR $r = 0.001$

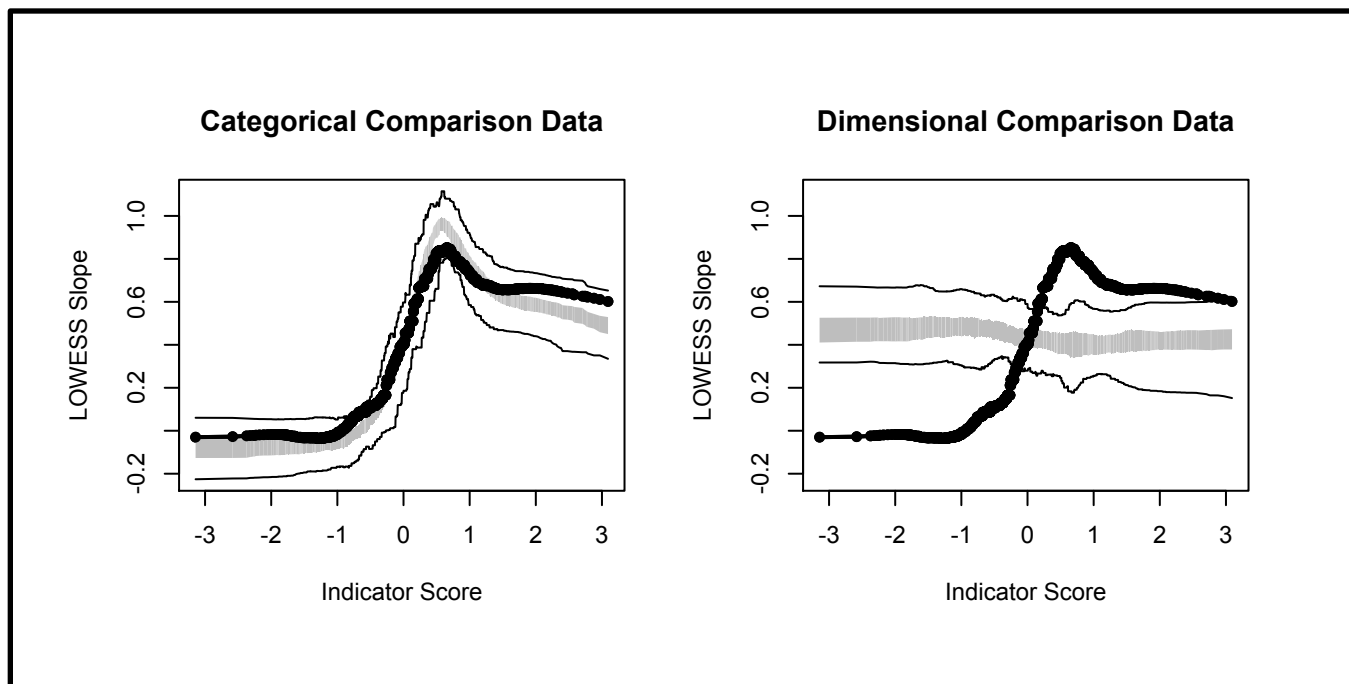
Analysis of 100 samples of categorical comparison data completed

Dimensional population generated, RMSR $r = 0$

Analysis of 100 samples of dimensional comparison data completed

Comparison curve fit index (CCFI) = $0.354 / (0.354 + 0.0648) = 0.845$

Note: CCFI values can range from 0 (dimensional) to 1 (categorical). The more a CCFI value deviates from .50, the stronger the result. When $.40 < \text{CCFI} < .60$, this should be interpreted with caution.



> MAXSLOPE(SkewedDim[,1:2])

SUMMARY OF MAXSLOPE ANALYTIC SPECIFICATIONS

Sample size: 600
Classification of cases: Cut total score at estimated base rate
Size of finite populations of comparison data: 1e+05
Number of samples of comparison data drawn from each population: 100

SUMMARY OF MAXSLOPE PARAMETER ESTIMATES

Estimated taxon base rate = 0.734

Indicator distributions in the full sample (N = 600):

	M	SD	Skew	Kurtosis
Ind 1	0.1	0.895	0.431	
Ind 2	0.1	0.858	0.323	
M	0.1	0.876	0.377	
SD	0.0	0.027	0.076	

Indicator distributions in the taxon (n = 440):

	M	SD	Skew	Kurtosis
Ind 1	0.308	0.974	0.733	0.207
Ind 2	0.360	0.917	0.849	0.419
M	0.334	0.945	0.791	0.313
SD	0.037	0.040	0.082	0.150

Indicator distributions in the complement (n = 160):

	M	SD	Skew	Kurtosis
Ind 1	-0.847	0.407	0.521	-1.750
Ind 2	-0.990	0.317	1.786	1.203
M	-0.919	0.362	1.153	-0.273
SD	0.101	0.064	0.894	2.089

Between-group validity:

	Raw Units	Cohen's d
Ind 1	1.155	1.343
Ind 2	1.350	1.682
M	1.253	1.513
SD	0.138	0.240

Indicator correlations:

Full Sample r = 0.396
Taxon r = 0.157
Complement r = -0.349

Generating and analyzing comparison data

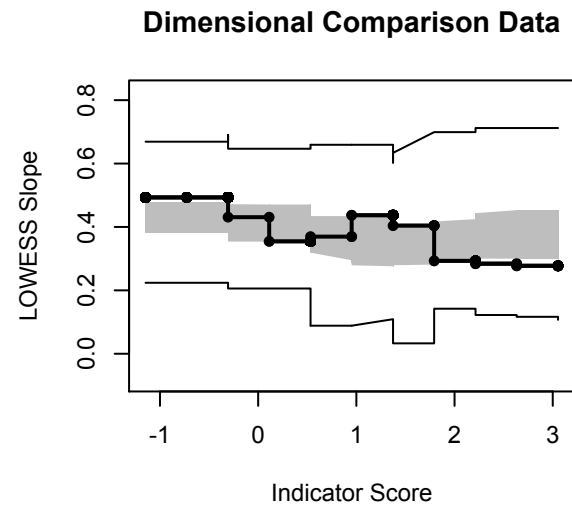
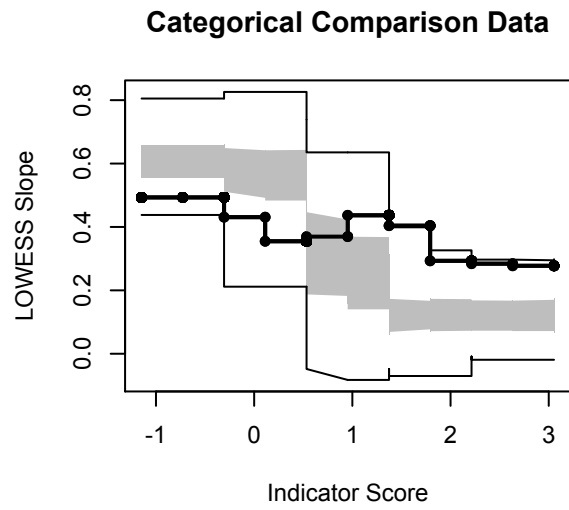
Categorical population generated, RMSR r = 0
Analysis of 100 samples of categorical comparison data completed

Dimensional population generated, RMSR r = 0
Analysis of 100 samples of dimensional comparison data completed

Comparison curve fit index (CCFI) = $0.056 / (0.056 + 0.1589) = 0.261$

Note: CCFI values can range from 0 (dimensional) to 1 (categorical). The more a CCFI value deviates from .50, the stronger the result. When $.40 < \text{CCFI} < .60$, this should be interpreted with caution.

[continued on the next page]



Chapter 11: P.Classify

Implementing the Base-Rate Classification Technique

11.1 Overview:

This program assigns cases to groups using the base-rate classification technique. Specifically, cases are sorted according to their total scores on all available indicators, and then the highest-scoring cases are assigned to the taxon such that the proportion of taxon members equals the specified base rate estimate. A study comparing two ways to classify cases using the results of MAXEIG analyses—Bayes' Theorem and the base-rate method—found that the latter performed better under all data conditions studied (Ruscio, 2009). Three ways of implementing the base-rate classification technique were evaluated: (1) using the base rate estimated by MAXEIG, (2) using the proportion of cases assigned to the taxon in the usual Bayesian manner as the base rate estimate, and (3) using the average of the base rates from (1) and (2) as the base rate estimate. For most conditions, the second method performed the best. A preprint of the accepted draft of this MS is available upon request.

The P.Classify program requires the provision of a data set and a base rate estimate; if the data set includes variables other than the indicators one wishes to use for classification, the columns containing the indicators must be specified. The program returns the same data file that was provided, with a column containing classification codes (1 = complement, 2 = taxon) appended at the end.

Note that if the base rate estimate locates the threshold dividing taxon and complement members at a score shared by many cases, all of these cases will be assigned to the same group. This may mean that the group sizes as classified do not precisely reproduce the specified taxon base rate. For example, if a total of 200 cases are to be assigned to the taxon and 190 score > 12, but the next 20 scores are tied at 12, this means that either all or none of these 20 cases will be assigned to the taxon, depending on which of these alternatives comes closest to reproducing the specified taxon base rate. If the threshold dividing taxon and complement members is identified by a tied score that equals the maximum score, then all cases at this score are assigned to the taxon (otherwise, all cases in the entire sample would be assigned to the complement); likewise, if the tied score equals the minimum score, all cases at this score are assigned to the complement.

11.2 Command:

```
P.Classify <- function(x, P.est, cols = 0)
```

11.3 Required Arguments:

`x` – The data set, which must include the indicators that will be used to classify cases and may include additional variables; if so, be sure to specify the columns that contain the indicators to use (see the optional argument `cols`, below).

`P.est` – The base rate estimate that will be used to classify cases

11.4 Optional Arguments:

`cols` – The columns within the data set that contain the indicators that will be used to classify cases. If this is not specified, the program assumes that all columns in the data set contain indicators, so it is important to specify the column numbers if this is not true. If the indicators appear in consecutive columns (e.g., 5 indicators appear in the 2nd through 6th columns of the data set), the simplest way to specify them would be as a range (e.g., `cols = 2:6`). If the indicators do not appear in consecutive columns (e.g., 5 indicators appear in the 2nd, 4th, 6th, 8th, and 10th columns), the simplest way to specify them would be as a list (e.g., `cols = c(2, 4, 6, 8, 10)`).

11.5 Running the Program:

Because this program returns the original data file with a new column appended, its output should be assigned to an object. One might wish to use the same object as input and output, but that is not necessary. The following command performs the base-rate classification technique on the `TestCat` data set (previously generated using `CreateData`) using a base rate estimate of .50 and saves the output in the original data set:

```
TestCat <- P.Classify(x = TestCat, P.est = .50, cols = 1:4)
```

Because all arguments are specified in the order that they appear in the function definition, this command could be simplified to:

```
TestCat <- P.Classify(TestCat, .50, 1:4)
```

If one did not want to append the classification codes to the original data set, the output could be assigned to a new data object:

```
TestCat.Class <- P.Classify(TestCat, .50, 1:4)
```

11.6 Output:

This program produces no text output; it returns a data object. As described above, the results of this program should be assigned to a data object—either the same one provided as input or a new one—for storage in the workspace. That is, rather than just calling the program, you name an object to hold the data that is generated. If you were to run the `P.Classify` program without assigning its results to an object, the output would be displayed in the console window but unavailable for subsequent analysis.

Chapter 12: Program Updates & References

History of Updates to the Program Code and References Cited in this Manual

Below is a dated listing of updates to the taxometric program code in the R language. For detailed information regarding each update described below, the pertinent chapter(s) within this manual are cited.

7/29/2014

- Graphical output for the MAMBAC, MAXEIG, L-Mode, and MAXSLOPE programs is restricted, by default, to the 2-panel plot that contains results for comparison data. Other graphs can be obtained by setting `Gr.Comp.Only = F` in these programs.
- Graph sheets intended for viewing, but not presenting (e.g., pages containing full panels of curves), are opened in a system-default window size and shape.

4/28/2014

- A rare problem with MAMBAC base rate estimation was addressed. The formula for estimating the base rate involves a division by the mean difference at the first cutting score, so if that value is 0 the program crashes. The solution implemented here is to examine this value and, as needed, advance to a cutting score that yields a nonzero value. Note that cutting scores at both ends are advanced together—if the lowest cutting score is moved inward by one case, so is the highest cutting score.
- The MAXEIG and MAMBAC programs no longer provide sampling distributions of base rate estimates for comparison data by default. The text output and graphs are available by setting `Gr.Base.Rates = T` in these programs (see Chs. 7 and 8).

2/26/2014

- Harold Kincaid's fix for Mac users was incorporated into the program code; it should have no effect on PC users.
- The CreateData program now allows users to specify a random number seed.

1/09/2012

- Because it does not appear that this is included in taxometric reports or that it has been supported by rigorous study, the GFI is no longer calculated or reported for any taxometric procedure.

11/13/2011

- The P.Classify program was modified such that cases with tied scores are always assigned to the same group. Previously, cases were assigned in strict accordance with the specified taxon base rate, and when this happened to locate the threshold for group assignment in the middle of a series of cases with tied scores, some of these cases were assigned to one group and some to the other (at random). Now, the program will check to determine whether this has occurred and, if so, reassign all cases at the tied score in question to the same group. For details on how this is done, see Chapter 11.

5/9/2011

- Graphs containing 2-panel plots with categorical and dimensional comparison data were reformatted. Rather than presenting the results for comparison data as the bounds extending ± 1 *SD*, a gray band contains the middle 50% of data points and outer lines show the minimum and maximum values. In other words, this new format shows the full range of values while highlighting the most common values.

4/22/2011

- In the program code, all instances of "win.graph" were replaced with "dev.new" to enhance compatibility across platforms. There should be no observable difference in the output on platforms that use the "win.graph" command. I am grateful to Franziska Borries for recommending this change.

7/26/2010

- In text and graphical output, the term "taxonic" was replaced with "categorical" to be more consistent with mainstream usage, as opposed to the more unique terminology in the taxometric literature. The same change in terminology was made throughout the text of this user's manual—in nearly 100 places!

- Two changes were made to the MAMBAC program. First, the option of running MAMBAC with cuts located at distinct scores on the input indicator was removed. It does not appear that this option was used very often and it will simplify future updates to remove this parameter. Second, the x axis labels were modified to indicate how many cuts were used. This makes the output more similar to that for the MAXEIG program, which indicates the number of windows (or intervals) that were used.

6/23/2010

- The MAXSLOPE procedure was reintroduced into the program code. To quote from the text below (6/12/2008), MAXSLOPE had been removed “for a variety of reasons (e.g., few investigators appear to be using it, no way to incorporate comparison data into the analysis has been developed, and the procedure has not been studied rigorously).” Ruscio and Walters (2011) introduced a way to calculate the CCFI for MAXSLOPE analyses, examined its performance across a wide range of data conditions, and found that this might be a useful adjunct to MAMBAC when only two indicators are available.

5/12/2010

- When running any of the taxometric procedures, missing data are now removed prior to analysis (listwise deletion). If there are any missing data, a note includes the numbers (and percentages) of cases retained and removed.

10/26/2009

- When comparison data are included in the analysis, results for L-Mode are now plotted in the same format as those for MAMBAC and MAXCOV/MAXEIG.

10/16/2009

- The MAMBAC and MAXEIG programs now check to see whether there are tied scores on any indicators, and if so they set the number of internal replications to a default value of 10 unless another value is specified (which could be 1, which amounts to no additional replication analyses). If no tied scores are found, the number of replications is set to 1 because additional analyses would increase run time with no change in results.
- The averaging of L-Mode curves for calculation of the CCFI was modified so that the averaged curve will be accurate even with very small sample sizes.
- The 3-panel plots of results for research and comparison data were eliminated because they are difficult to read when a large number of comparison data sets are analyzed—which is now feasible and strongly recommended.
- The graphs showing the distributions of the Ms and SDs of taxon base rate estimates were eliminated because it appears that the numerical summary in the text output is reported much more often.
- The option of using SD intervals in a MAXCOV analysis was eliminated because it appears to be used seldom, if ever, and it does not allow the averaging of curves as required to calculate the CCFI.

8/3/2009

- The L-Mode classification of cases was corrected such that when “Save.Class = T,” class assignment codes now are returned in the same order as the cases appeared in the original data set (see Chapter 9 for details).

7/16/2009

- By default, the L-Mode program now standardizes indicators prior to analysis to equate their variances. Users can override this setting (see Chapter 9 for details).

5/11/2009

- For MAXCOV and MAXEIG analyses, the default number of windows was changed from 50 to 25. This change was based on a simulation study that found, among other things, that using a small number of windows was the most effective way to divide cases into subsamples for these procedures (Walters & Ruscio, 2010).

1/19/2009

- A glitch in the option for saving classification codes generated by the MAXEIG program was repaired.

11/24/2008

- A new parameter was added to each taxometric procedure such that the user can specify a random number seed. This value is set prior to analysis of the empirical data as well as prior to generating each population of comparison data. There are two benefits to this modification. First, by controlling random number seeds one can obtain exact replications of results that involve a random element in the analysis (e.g., when internal replications are used). Second, by providing the same seed across taxometric procedures, one can analyze identical populations and samples of comparison data.

11/22/2008

- Within each taxometric procedure, the same assignment of cases to groups is used to estimate data parameters and to generate taxonomic comparison data (if comparison data are requested). Previously, data parameters might be estimated for one classification (e.g., base-rate classification using the estimated taxon base rate) with taxonomic comparison data generated on the basis of another classification (e.g., one supplied by the user). The summary that appears at the top of the text output now indicates what technique is used for both purposes.

6/12/2008

- The calculation of CCFI values for L-Mode analyses was updated; see Chapter 9 for details. Research in progress suggests that this differentiates between taxonomic and dimensional data at least accurately as the CCFI calculated from MAMBAC, MAXCOV, or MAXEIG results. Also, the factor score density plots for the research and comparison data are now provided in a single graph.
- Averaged graphs for MAMBAC, MAXCOV, and MAXEIG now appear in smaller windows that no longer need to be cropped.
- MAXSLOPE was removed from the program file for a variety of reasons (e.g., few investigators appear to be using it, no way to incorporate comparison data into the analysis has been developed, and the procedure has not been studied rigorously).
- Options to perform the inchworm consistency test were removed from the MAXEIG program. This test remains highly recommended whenever a cusped curve emerges and/or the putative taxon base rate is very low (or very high). To implement the test, the MAXEIG program should be run multiple times, with different numbers of windows specified each time. These program options were dropped because they made updating and checking the software unnecessarily cumbersome.

4/5/2008

- Parameters in the CreateData program were renamed to be clearer and/or more consistent with common usage in the taxometrics literature. “Type” became “Str” for taxonomic or dimensional data, “Ind” became “k” for the number of indicator variables, “Sep” became “d” for indicator validity, and “Corr” became “r” for the correlation between indicators in dimensional data.

2/12/2008

- With the changes on 1/21/2008, the default number of comparison data sets became 100 for each structure. Whereas previously the *M* and *SD* of base rate estimates for each sample of comparison data was printed (for MAMBAC, MAXCOV, and MAXEIG), now only a summary of these values is printed.

1/21/2008

- For MAMBAC, MAXCOV, MAXEIG, and L-Mode analyses, the default option was changed to including comparison data. Now that a number of published studies have supported the utility of this approach, it has become the default. Users can override this by using “Comp.Data = F” rather than the default of “Comp.Data = T” when running the programs.
- Based on developments described in Ruscio and Kaczetow (2008), the algorithm for generating the comparison data was updated. Formerly, the programs DimSample and TaxSample were used to generate each requested sample of comparison data. Now, the GenData program is used to generate a finite population of cases for each structure from which multiple samples are drawn at random. This yields several benefits.

First, given the large size of the population being generated (program default = 100,000), the correlations in this population usually can reproduce those in the original data set very accurately.

Second, when each sample of comparison data is randomly sampled from its population, all characteristics will vary according to normal sampling error.

Third, it is now easier to generate and analyze larger numbers of samples of comparison data. The DimSample and TaxSample programs were run once per sample, and that could be very time consuming. The GenData program is only run to generate the population of taxonic data and the population of dimensional data. Drawing random samples from these populations requires almost no time at all, so once the populations are generated drawing a large number of samples is feasible. Because of this, the program default has been changed to 100 samples of comparison data per structure.

- To indicate that analyses were performed using the new technique and the GenData program, plots are now labeled “Taxonic Comparison Data” and “Dimensional Comparison Data.” Analyses performed using the previous technique, with DimSample and TaxSample, were labeled “Simulated Taxonic Data” and “Simulated Dimensional Data.”
- The CCFI is now provided for L-Mode analyses. The calculation of Fit_{RMSR} values was adapted for L-Mode by using minimal Euclidean distances rather than ordinal residuals (see Chapter 4). Please note that whereas a number of studies (published and unpublished) have tested the performance of the CCFI when calculated from the results of MAMBAC, MAXCOV, and MAXEIG analyses, its performance has not been tested rigorously for L-Mode analyses. Preliminary examination suggests that it works well, and this use of the index warrants further study.

1/20/2008

- A few precautions were added to ensure that comparison data can be generated when requested and to prevent the programs from crashing under certain circumstances.

First, when base-rate classification is used to assign cases to groups prior to generating taxonic comparison data, the minimal number of cases in each group is set to 20. This prevents the program from crashing when a procedure estimates the base rate at 0 or 1, and it ensures that comparison data are generated at least this minimal group size.

Second, the program checks to ensure that there is variance on each indicator within each group when generating comparison data or when estimating latent parameters. If all cases assigned to a group happen to score the same on a particular indicator, a very small amount of variation is induced by adding random normal values drawn from a population with $\mu = 0$, $\sigma = .0001$. This has a negligible effect on between-group separation when calculated in raw units (the standardized measure d cannot be calculated when the $SD = 0$). In addition, after having drawn a bootstrap sample for an indicator, the program checks once again that this specific distribution possesses variance. If not, a new bootstrap sample is drawn. What these checks prevent is a situation in which an entirely or nearly homogeneous distribution yields a bootstrap sample with no variance, in which case the program crashes when it attempts to calculate the target correlation matrix to reproduce—this requires all $SD > 0$.

Third, the default classification technique for MAXCOV/MAXEIG analyses was changed from Bayes’ Theorem to base-rate classification. This was done for two reasons: Using Bayes’ Theorem sometimes causes the program to crash (e.g., when the estimated VP and FP rates are at the extremes of 0 and 1), and a Monte Carlo study suggests that base-rate classification attains greater classification accuracy across a wide range of data conditions.

6/9/2007

- The DimSample program, which generates dimensional comparison data (see Chapter 4), was revised in two ways.

First, the program used to contain a loop over all cases in the data set nested within a loop over all indicators. The former loop was removed, which improves the speed of the program, especially with large samples. This change has no effect on the output of the program, it simply accelerates processing.

Second, the criterion to determine the number of factors for reproducing the indicator correlation matrix was changed. Previously, the program used the liberal Kaiser criterion of the number of eigenvalues > 1 . Now, a parallel analysis is performed using 100 random data sets (see Reise, Waller, & Comrey, 2000).

2/4/2007

- The P.Classify program was added to perform the base-rate classification technique using any base rate estimate.

12/19/2006

- The algorithm used to calculate Bayesian probabilities of taxon membership in a MAXCOV or MAXEIG analysis was modified to run considerably faster. Results are not affected by this change.

6/3/2006

- In a MAXCOV or MAXEIG analysis, one can now save either the classification of cases (1 = taxon, 2 = complement) or the Bayesian probability of taxon membership for each case. Previously, only the former option was available. I am grateful to Bobbi Carothers for providing the modifications to make the latter option available. See Chapter 8 for details and additional information on performing MAXEIG or MAXCOV analyses.

2/24/2006

- In the text summary for the output of the MAXEIG program, the calculation method (covariances vs. eigenvalues) is no longer specified. Instead, whenever one uses covariances all output is labeled as MAXCOV and whenever one uses eigenvalues all output is labeled as MAXEIG. See Chapter 8 for information on performing MAXEIG or MAXCOV analyses.
- The Validity.Est program was revised and renamed Indicator.Dist to better reflect the information that it provides (see Chapter 5). Specifically, all four distributional moments (mean, variance [actually, SD is used rather than variance because this is customary in the taxometrics literature], skew, and kurtosis) are calculated for the full sample of data and within groups (taxon and complement) identified by the user. This is followed by estimates of indicator validity (between-group separation, in both raw and standardized [Cohen's d] form) and indicator correlation matrices calculated in the full sample and within groups.

1/26/2006

- The output for curve fit index was modified. Rather than providing the CCFI alone, its two components are shown as well (see Chapter 4 for the calculation of the CCFI).

8/21/2005

- The calculation of the curve fit index, which was updated on 8/12/2005, was revised slightly. The new index, called the Comparison Curve Fit Index (CCFI) is described in Chapter 4.

8/12/2005

- The programs used to generate taxonic and dimensional comparison data were revised in a number of important ways (see Chapter 4 for details). Also, the calculation of a curve fit index for MAMBAC, MAXCOV, and MAXEIG analyses was revised.

12/15/2004

- When R reached version 2.x, a glitch arose in the way the programs checked version numbers to determine whether to use the "eda" or "stats" library for certain functions, such as smoothing. This was repaired.

7/16/2004

- The assignment of cases to classes using Bayes' Theorem in the MAXEIG program was modified to prevent a computational problem that arises when the VP and FP rates are both estimated to be 1.00 (when the hitmax is located in the lowest subsample). When this occurs, a division by 0 occurs in Bayes' Theorem. The solution is not to use any indicator whose hitmax is located in the lowest subsample when calculating the probability of taxon membership (see Chapter 8 for more on using the MAXEIG program).

7/9/2004

- The CreateData program was modified to allow the generation of indicators with negative skew (see Chapter 3).

7/6/2004

- An error in the calculation of GFI values was fixed. This applies to GFI values calculated using the MAXEIG, MAMBAC, MAXSLOPE, and L-Mode programs. *All GFIs calculated with prior versions of these programs were calculated incorrectly and should be recalculated using the corrected programs.*
- When using SD units to define intervals in a MAXEIG or MAXCOV analysis, they are now determined by beginning at $x = 0$ (along the standardized input indicator) and working outward in each direction until the specified minimal subsample size requires collapsing the remaining cases into a single interval. Previously, the program worked inward from the extremes, collapsing until the minimal subsample size was achieved at each end and then leaving all intermediate intervals intact (even if they fell below the minimal subsample size used at the extremes). The new technique results in a number of intervals less than or equal to that of the previous technique, but the minimal subsample size is maintained within every interval (see Chapter 8 for more on using the MAXEIG program).

6/23/2004

- Stand-alone functions to compute skew and kurtosis were added (see Chapters 5 and 6 for details).

6/19/2004

- As of version 1.9, the supplementary R code that includes Tukey's running medians smoother is in the "stats" package rather than the "eda" package. By checking to see which version of R you are running, the MAXEIG and L-Mode programs should now request the appropriate package. My thanks to James Prisciandaro for pointing out the problem and its solution.

6/11/2004

- Corrected a problem in the L-Mode program that sometimes caused the plot to be reversed along the x axis. Specifically, when the factor analysis program returns negative (rather than positive) loadings for all indicators, the computed factor scores are now multiplied by -1 to fix this problem (see Chapter 9 for more on using the L-Mode program).

4/2/2004

- Modified the CreateData program such that within-group correlations are specified separately for the taxon and complement (see Chapter 3).
- Removed the post-hoc estimation of indicator validity using the formula provided by Meehl & Yonce (1996, p. 1146) from the Validity.Est program as well as each taxometric procedure. This was done because more direct estimates of indicator validity are available using classified cases (see Chapter 5).

3/16/2004

- Compatibility with R version 1.8.1 was checked when I upgraded. A warning message that used to appear for each sample of comparison data was fixed. (For those using previous versions of the taxometric programs, this warning message can safely be ignored.)

3/2/2004

- Sampling distributions of taxon base rate estimates are now generated when you request that multiple samples of comparison data be generated. These results are presented in the text output (in between the sections that report the accuracy with which correlations were reproduced and the fit of averaged curves) and in a new graph. Details and examples appear in Chapters 7 and 8.
- When generating taxonic comparison data using the base-rate classification technique in the MAMBAC or MAXEIG program, the M base rate estimate observed in analyses of the research data is used to assign cases to putative taxon and complement classes rather than the base rate estimate from the averaged curve (see Chapters 4, 7, and 8).

References

The following sources were cited in this manual. For additional references on the taxometric method, including many published studies that have used the method, see the “Taxometric References” file at www.tcnj.edu/~ruscio/taxometrics.html.

- Beach, S. R. H., Amir, N., & Bau, J. J. (2005). Can sample-specific simulations help detect low base-rate taxonicity? *Psychological Assessment, 17*, 446-461.
- Cleveland, W. S. (1979). Robust locally-weighted regression and smoothing scatterplots. *Journal of the American Statistical Association, 74*, 829-836.
- Grove, W. M. (2004). The MAXSLOPE taxometric procedure: Mathematical derivation, parameter estimation, consistency tests. *Psychological Reports, 95*, 517-550.
- Grove, W. M., & Meehl, P. E. (1993). Simple regression-based procedures for taxometric investigations. *Psychological Reports, 73*, 707-737.
- Meehl, P. E., & Yonce, L. J. (1994). Taxometric analysis: I. Detecting taxonicity with two quantitative indicators using means above and below a sliding cut (MAMBAC procedure). *Psychological Reports, 74*, 1059-1274.
- Meehl, P. E., & Yonce, L. J. (1996). Taxometric analysis: II. Detecting taxonicity using covariance of two quantitative indicators in successive intervals of a third indicator (MAXCOV procedure). *Psychological Reports, 78*, 1091-1227.
- Reise, S. P., Waller, N. G., & Comrey, A. L. (2000). Factor analysis and scale revision. *Psychological Assessment, 12*, 287-297.
- Ruscio, J. (2007). Taxometric analysis: An empirically-grounded approach to implementing the method. *Criminal Justice and Behavior, 24*(12), 1588-1622.
- Ruscio, J. (2009). Assigning cases to groups using taxometric results: An empirical comparison of classification techniques. *Assessment, 16*(1), 55-70.
- Ruscio, J., Haslam, N., & Ruscio, A. M. (2006). *Introduction to the taxometric method: A practical guide*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Ruscio, J., & Kacetow, W. (2008). Simulating multivariate nonnormal data using an iterative algorithm. *Multivariate Behavioral Research, 43*, 355-381.
- Ruscio, J., & Kacetow, W. (2009). Differentiating categories and dimensions: Evaluating the robustness of taxometric analysis. *Multivariate Behavioral Research, 44*, 259-280.
- Ruscio, J., & Marcus, D. K. (2007). Detecting small taxa using simulated comparison data: A reanalysis of Beach, Amir, and Bau's (2005) data. *Psychological Assessment, 19*, 241-246.
- Ruscio, J., & Ruscio, A. M., & Keane, T. M. (2004). Using taxometric analysis to distinguish a small latent taxon from a latent dimension with positively skewed indicators: The case of Involuntary Defeat Syndrome. *Journal of Abnormal Psychology, 113*, 145-154.
- Ruscio, J., Ruscio, A. M., & Meron, M. (2007). Applying the bootstrap to taxometric analysis: Generating empirical sampling distributions to help interpret results. *Multivariate Behavioral Research, 44*, 349-386.
- Ruscio, J., & Walters, G. D. (2009). Using comparison data to differentiate categorical and dimensional data by examining factor score distributions: Resolving the mode problem. *Psychological Assessment, 21*, 578-594.
- Ruscio, J., & Walters, G. D. (2011). Differentiating categorical and dimensional data with taxometric analysis: Are two variables better than none? *Psychological Assessment, 23*, 287-299.
- Ruscio, J., Walters, G. D., Marcus, D. K., & Kacetow, W. (2010). Comparing the relative fit of categorical and dimensional latent variable models using consistency tests. *Psychological Assessment, 22*, 5-21.
- Waller, N. G., & Meehl, P. E. (1998). *Multivariate taxometric procedures: Distinguishing types from continua*. Thousand Oaks, CA: Sage.
- Walters, G. D., & Ruscio, J. (2009). To sum or not to sum: Taxometric analysis with ordered categorical assessment items. *Psychological Assessment, 21*, 99-111.
- Walters, G. D., & Ruscio, J. (2010). Where do we draw the line? Assigning cases to subsamples for MAMBAC, MAXCOV, and MAXEIG taxometric analyses. *Assessment, 17*, 321-333.